Proceedings of the ECAI Workshop on Formal Concept Analysis for Artificial Intelligence (FCA4AI)

Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph

Workshop Notes

International Workshop

"What can FCA do for Artificial Intelligence?"

FCA4AI

August 28, 2012

Montpellier, France

held at the

European Conference on Artificial Intelligence 2012

Editors

Sergei O. Kuznetsov (NRU HSE Moscow)

Amedeo Napoli (LORIA Nancy)

Sebastian Rudolph (AIFB KIT Karlsruhe)

http://www.fca4ai.hse.ru

What FCA Can Do for Artificial Intelligence?
FCA4AI: An International Workshop

Preface

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications) which can be used for many AI needs, e.g. knowledge processing involving learning, knowledge discovery, knowledge representation and reasoning, ontology engineering, as well as information retrieval and text processing. Thus, there exist many "natural links" between FCA and AI.

Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing, such as work on pattern structures and relational context analysis. These extensions are aimed at allowing FCA to deal with more complex than just binary data, both from the data analysis and knowledge discovery points of view and from the knowledge representation point of view, including, e.g., ontology engineering.

All these works extend the capabilities of FCA and offer new possibilities for AI activities in the framework of FCA. Accordingly, in this workshop, we are interested in two main issues:

- How can FCA support AI activities such as knowledge processing (knowledge discovery, knowledge representation and reasoning), learning (clustering, pattern and data mining), natural language processing, information retrieval.
- How can FCA be extended in order to help AI researchers to solve new and complex problems in their domains.

The workshop is dedicated to discuss such issues. The papers submitted to the workshop were carefully peer-reviewed by two members of the program committee and 11 papers with the highest scores were selected. We thank all the PC members for their reviews and all the authors for their contributions. We also thank the organizing committee of ECAI-2012 and especially workshop chairs Jérôme Lang and Michèle Sebag for the support of the workshop.

The Workshop Chairs

Sergei O. Kuznetsov
National Research University Higher Schools of Economics, Moscow, Russia

Amedeo Napoli
LORIA (CNRS – INRIA – Université de Lorraine), Vandoeuvre les Nancy, France

Sebastian Rudolph
AIFB Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Program Committee

Mathieu D'Aquin (Open University, UK)

Franz Baader (Technische Universität Dresden, Germany)

Radim Bělohlávek (Palacký University, Olomouc, Czech Republic)

Claudio Carpineto (Fondazione Ugo Bordoni, Roma, Italy)

Felix Distel (Technische Universität Dresden, Germany)

Sébastien Ferré (IRISA Rennes, France)

Bernhard Ganter (Technische Universität Dresden, Germany)

Pascal Hitzler (Wright State University, Dayton, Ohio, USA)

Marianne Huchard (LIRMM Montpellier, France)

Dmitry I. Ignatov (NRU Higher School of Economics, Moscow, Russia)

Mehdi Kaytoue (Universidade Federal Minas Gerais, Belo Horizonte, Brazil)

Markus Krötzsch (University of Oxford, UK)

Sergei A. Obiedkov (NRU Higher School of Economics, Moscow, Russia)

Uta Priss (Ostfalia University of Applied Sciences, Wolfenbüttel, Germany)

Baris Sertkaya (SAP Dresden, Germany)

Gerd Stumme (Universität Kassel, Germany)

Petko Valtchev (Université du Québec à Montréal, Montréal, Canada)

Table of Contents

Invited Talk

Relational Concept Analysis: a synthesis and open questions

Marianne Huchard
LIRMM, Université de Montpellier 2 and CNRS, Montpellier, France,
marianne.huchard@lirmm.fr

Abstract

Relational Concept Analysis (RCA) builds conceptual structures on sets of objects connected by sets of links, following an underlying entity-relationship diagram. These conceptual structures (concept lattice families) are composed of several concept lattices (one for each object set one wants to focus on) connected by relational attributes of various strengths. Concept lattice families can be read to extract interconnected relevant object groups and classifications as well as to derive implication rules. The RCA algorithm uses classical concept lattice building algorithms and a relational scaling step. In this talk, we recall the main principles of RCA and we elaborate on several issues (some of which are totally open) including querying relational data with RCA, looking at specific relational schemes, convergence of RCA when disturbing the classical algorithmic schema, and understanding the growth process of concepts.

Formal Concept Analysis Applied to Transcriptomic Data

Mehwish Alam[2,3], Adrien Coulet[2,3], Amedeo Napoli[1,2], Malika Smaïl-Tabbone[2,3]

[1] CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
[2] Inria, Villers-lès-Nancy, F-54600, France
[3] Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
{mehwish.alam,adrien.coulet,amedeo.napoli,malika.smail}@inria.fr

Abstract. Identifying functions or pathways shared by genes responsible for cancer is still a challenging task. This paper describes the preparation work for applying Formal Concept Analysis (FCA) to biological data. After gene transcription experiments, we integrate various annotations of selected genes in a database along with relevant domain knowledge. The database subsequently allows to build formal contexts in a flexible way. We present here a preliminary experiment using these data on a core context with the addition of domain knowledge by context apposition. The resulting concept lattices are pruned and we discuss some interesting concepts. Our study shows how data integration and FCA can help the domain expert in the exploration of complex data.

Keywords: Formal Concept Analysis, Knowledge Discovery, Data Integration, Transcriptomic Data.

1 Introduction

Over past few years, large volumes of transcriptomic data were produced but their analysis remains a challenging task because of the complexity of the biological background. In the field of transcriptomics, biologists analyze routinely the transcription or expression of genes in various situations (e.g., in tumor samples versus non-tumor samples).

Some earlier studies aimed at retrieving sets of genes sharing the same transcriptionl behaviour with the help of Formal Concept Analysis (see, e.g., [7, 10, 11]). Further studies analyze gene expression data by using gene annotations to determine whether a set of differentially expressed genes is enriched with biological attributes [1, 2, 13]. Many useful resources are available online and several efforts have been made for integrating heterogeneous data [5, 8]. A recent example is of the Broad Institute where biological data were gathered from multiple resources to get thousands of predefined gene sets stored in the Molecular Signature DataBase, MSigDB [4]. A predefined gene set is a set of genes known to have a specific property such as their position on the genome, their involvement in a

biological process (or a molecular pathway) etc. Subsequently, given an experimental gene list as input the GSEA (Gene Set Enrichment Analysis) program is used to asses whether each predefined gene set (in the MSigDB database) is significantly present in the input list by computing an enrichment score [3].

In this paper, we are interested in applying knowledge discovery techniques for analyzing a differentially expressed gene set and identifying functions or pathways shared by these genes assumed to be responsible for cancer. Knowledge discovery aims at extracting relevant and useful knowledge patterns from a large amount of data. It is an interactive and iterative process involving a human (analyst or domain expert) and data sources. We show how various gene annotations and domain knowledge are integrated in a database which is then queried for building in a flexible way formal contexts. We present here a preliminary experiments using these data. It was performed on a core context with the addition of domain knowledge (by context apposition). The considered domain knowledge are the hierarchical relationships between molecular pathways. Pruning the obtained lattices allows us to retrieve interesting concepts which we discuss. The results obtained from both experiments are also compared.

The plan of the paper is as follows: Section 2 introduces Formal Concept Analysis, Section 3 explains the data resources which are integrated, Section 4 focuses on the application of FCA, Section 5 discusses the results and Section 6 concludes the paper and presents future Work.

2 Formal Concept Analysis

We introduce here the basics of Formal Concept Analysis that are needed to understand what follows. Let G and M be the set of objects and set of attributes respectively and I be the relation between the objects and the attributes $I \subseteq G \times M$, where $g \in G$, $m \in M$, gIm is true iff the object g has the attribute m. The triple $K = (G, M, I)$ is called a formal context. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then a Galois connection denoted by $'$ is given by:

$$A' := \{m \in M \mid gIm \; \forall \; g \in A\} \tag{1}$$

$$B' := \{g \in G \mid g \; I \; m \; \forall \; m \in B\} \tag{2}$$

FCA framework is fully described in [6]. FCA helps in defining concepts which are composed of a maximal set of objects sharing a maximal set of attributes. However, given an input context, the resulting concept lattice can be very large leading to computational and interpretation problems. In order to have reduced and meaningful concepts, one can select concepts whose support is greater than a certain threshold, i.e., the iceberg lattice. For a concept (A,B), the support is the cardinality of the extent A. An alternative is to use the notion of stability that was proposed in [9, 12]. The stability index measures how much the concept intent depends on particular objects of the extent.

3 Complex Biological Data Integration

In this section, we introduce and describe the biological data on which we are working.

3.1 Molecular Signature Database (MSigDB)

Molecular Signature Database (MSigDB) is an up-to-date database which contains data from several resources such as KEGG, BIOCARTA, REACTOME, and Amigo [4]. It is a collection of 6769 predefined gene sets. A predefined gene set is a set of genes having a specific property such as their position on the genome (e.g., the genes at position chr5q12, i.e., band 12 on arm q of chromosome 5), their involvment in a biological process or a molecular pathway (e.g., the genes which are involved in the KEGG APOPTOSIS pathway)... A pathway is a series of actions among molecules in a cell that leads to a certain change in a cell. KEGG is a database storing hundreds of known pathways[4]. Besides, the MSigDB gene sets are grouped into five categories (Table 1). For instance, all the gene sets which are defined on the basis of gene position belong to the category C1. The category C5 groups the gene sets defined on Gene Ontology (GO) terms annotating the genes (with respect to their molecular function or their housing cellular component).

For our study, we used MSigDB Version 3.0. One entry, shown below in XML format, describes the gene set corresponding to the GO term 'RNA Polymerase II Transcription Factor Activity Enhancer Binding' (all the attribute names are underlined). The *Members* attribute contains the list of gene symbols belonging to the gene set. MSigDB was chosen as the main source for describing genes because it gathers up-to-date informations about many aspects of human genes.

<GENESET Standard Name ="RNA Polymerase II Transcription Factor Activity Enhancer Binding" Systematic Name = "M900" Historical Names ="" Organism ="Homo sapiens" Geneset Listing URL ="" Chip = "Human Gene Symbol" Category Code ="c5" Sub Category Code ="MF" Contributor ="Gene Ontology" Contributor Org ="GO" Description Brief ="Genes annotated by the GO term GO:0003705. Functions to initiate or regulate RNA polymerase II transcription by binding an enhancer region of DNA." Description Full ="" Members =" MYOD1, TFAP4, EPAS1, RELA, MYF5, MYEF2, NFIX, PURA, HIF1A" Members Symbolized = "MYOD1, TFAP4, EPAS1, RELA, MYF5, MYEF2, NFIX, PURA, HIF1A" Members EZID =" 7023, 2034, 5970, 3091" Members Mapping = " MYOD1, 4654-TFAP4, TFAP4, 7023-EPAS1, EPAS1, 2034-RELA, RELA, 5970-MYF5, MYF5, 4617-MYEF2, MYEF2, 50804-NFIX, NFIX, 4784-PURA, PURA, 5813-HIF1A" Status ="public" > </GENESET>

3.2 Domain Knowledge

Besides the gene annotations included in MSigDB, many types of domain knowledge are interesting to use when analyzing genes. The first type of such do-

[4] http://www.genome.jp/kegg/pathway.html

Table 1. Categories of MSigDB Gene Sets

Category	Description	Data Provenance
C1: Positional Gene Sets	Location of the gene on the chromosome.	Broad Institute
C2: Curated Gene Sets	Pathways	KEGG, REACTOME, BIOCARTA
C3: Motif Gene Sets	microRNAs, Transcription Factor Targets.	Broad Institute
C4: Computational Gene Sets	Cancer Modules	Broad Institute
C5: Gene Ontology (GO) Gene Sets	Biological Process, Cellular Components, Molecular Functions	AmiGO

main knowledge are the hierarchical relationships between GO terms or between KEGG pathways. Indeed, the KEGG hierarchy for human groups the KEGG pathways into 40 categories and 6 upper level categories. Figure 1 illustrates the KEGG hierarchy detailing on one upper-level category and one category.

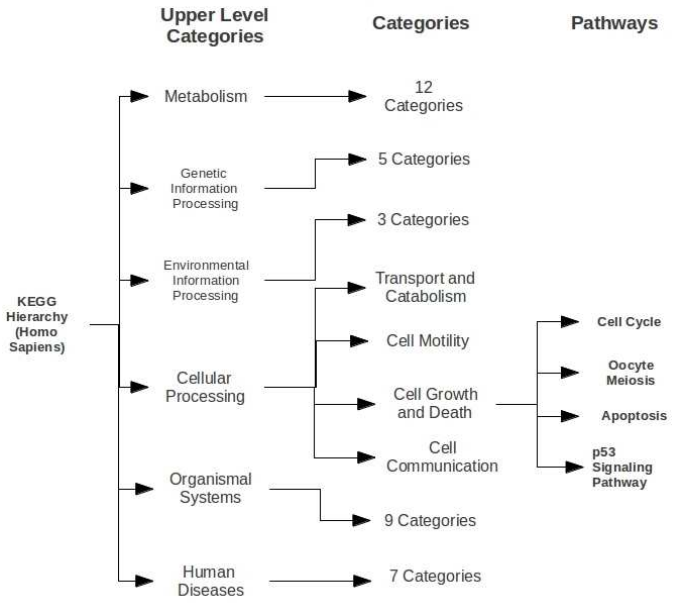


Fig. 1. Hierarchical Relationship in KEGG

In our study we have genes described by pathways involving them which may in turn be present in some category of pathways. For example, if a gene is involved in a pathway apoptosis it will also be in the category 'Cell Growth and Death'. In order to facilitate the knowledge discovery, it is important to

identify the relevant data sources, organize, and integrate the data at one single database. In our case, the relevant primary data sources are MSigDB, KEGG PATHWAYS database, and AmiGO database.

4 From Data to Knowledge

Once the data are integrated in our database the next step is to build formal contexts for applying FCA. Our experiment focuses on applying FCA to a core context describing genes by MSigDB-based attributes and shows its extension based on the addition of domain knowledge.

4.1 Test Data Sets

The experiments described here are based on three published sets of genes corresponding to Cancer Modules defined in [14]. The authors compiled gene sets from various resources and a large collection of micro-array data related to cancers. These modules correspond to gene sets whose expression significantly change in a variety of cancer conditions (they are also defined as MSigDB gene sets in the C4 category). Our test data are composed of three lists of genes corresponding to the Cancer Modules 1 (Ovary Genes), 2 (Dorsal Root Ganglia Genes), and 5 (Lung Genes).

4.2 Using FCA for Analyzing Genes

We apply FCA for analyzing a context describing genes of each Cancer Module with MSigDB-based attributes. Table 2 shows five genes (involved in Cancer Module 1) as a set of objects described by attributes which are the memberships to gene sets from MSigDB. For example, CCT6A is in the set of genes (gene set) whose standard_name is *Reactome Serotonin Receptors*. Interestingly, by querying our integrated database the analyst is able to select the predefined gene sets to include in the formal context.

In order to extend the analysis of a list of genes, we need to take into account the domain knowledge. Hence, the same experiment was conducted with the addition of the KEGG hierarchy knowledge to the core contexts resulting in three extended contexts. All KEGG categories and upper-level categories were added as a set of attributes. If a gene is member of a KEGG pathway which in turn belongs to a category and an upper level category then a cross '$\times$' is added in the corresponding cells in the extended context.

Table 2 shows five genes (from Cancer Module 1) with the addition of one KEGG category (kc) and one KEGG upper level category (kuc). In the given example *CCT6A* is involved in pathway *KEGG PPAR Signaling Pathway* which belongs to the category *kc:Endocrine System* and upper level category *kuc:Organismal Systems*. The lattices were generated and the statistics for each Cancer Module are given in Table 3. The concepts were filtered and ranked based on same criteria as in the first experiment.

Table 2. A Toy Example of Formal Context with Domain Knowledge

Genes	TTTGCAC,MIR-19A,MIR-19B	Reactome Serotonin Receptors	KEGG PPAR Signaling Pathway	V$POU3F2_02	GO Cellular Component Assembly	chr5q12	kc:Endocrine System	kuc:Organismal Systems
BTB03	×				×	×		
PSPHL			×	×			×	×
CCT6A		×				×		
QNGPT1	×	×			×	×		
MYC	×			×				

Table 3. Concept Lattice Statistics for the Cancer Modules with Domain Knowledge

Data Sets	No. of Genes	No. of Attributes	No. of Concepts	Levels
Module 1	361	3496	9,588	12
Module 2	378	3496	6,508	11
Module 5	419	3496	5,004	12

5 Results

In this study, biologists are interested in links between the input genes in terms of pathways in which they participate, relationship between genes and microRNAs etc. We obtained concepts with shared transcription factors, pathways, positions of genes and some GO terms. After the selection of concepts with higher support, we observed that there were some concepts with pathways from KEGG and REACTOME as their intent. These pathways are either related to cell proliferation or apoptosis (cell death). The addition of domain knowledge effectively gives an opportunity to obtain the pathway categories shared by larger sets of of genes. Table 4 shows the top-ranked concepts found in each module. For example, in module 5, we have confirmation that *Cytokine Cytokine Receptor Interaction* pathway comes under the category *Signaling Molecules and Interaction* and upper level category *Environmental Information Processing* (Concept ID 4938). The absolute support and stability of the concept containing only the category *Signaling Molecules and Interaction* and upper level category *Environmental Information Processing* as its intent are higher (Concept ID 4995, Table 4) .

To sum up, we were able to discover interesting biological properties of subsets of genes in the three test data sets. As for example, the Focal Adhesion pathway was found to be associated to 17 genes in both modules 1 and 2; the

KEGG category Immune System was found to be shared by 11 to 25 genes in the three cancer modules (Table 4). Given the test data sets, these results are hopeful and constitute interesting positive control. This confirms that FCA-based analysis offers a powerful procedure to deeply explore sets of genes.

Table 4. Top-ranked Concepts with Domain Knowledge

Dataset	Concept ID	Intents	Absolute Support	Stability
Module 1	**9585**	M2192:GGGAGGRR _V$MAZ_Q6	51	0.99
	9571	M2598:GO Membrane Part	27	0.99
	9566	kc:Immune System, kuc:Organismal Systems	25	0.99
	9402	chr19q13	10	0.99
	9078	M10792:KEGG MAPK Signaling Pathway, kc:Signal Transduction, kuc:Environmental Information Processing	12	0.87
Module 2	**6502**	M2192:GGGAGGRR _V$MAZ_Q6	44	0.99
	6496	kc:Immune System, kuc:Organismal Systems	15	0.99
	6388	chr6p21	10	0.97
	6335	M10792:KEGG MAPK Signaling Pathway, kc:Signal Transduction, kuc:Environmental Information Processing	11	0.89
Module 5	**5002**	kuc:Cellular Processes	48	0.99
	4995	kc:Signaling Molecules and Interaction, kuc:Environmental Information Processing	26	0.99
	4933	chr19q13	11	0.99
	4985	kc:Immune System, kuc:Organismal Systems	11	0.99
	4938	M9809:KEGG Cytokine Cytokine Receptor Interaction, kc:Signaling Molecules and Interaction, kuc:Environmental Information Processing	11	0.87

6 Conclusion and Future Work

Our study shows how Formal Concept Analysis can be applied to complex biological data. Data integration and FCA give the flexibility of using various types of attributes (pathways, GO terms, positions, microRNAs and Transcription Factor Targets) for analyzing a list of genes. Our approach gives an insight into how domain knowledge can be introduced in the analysis with the help of

FCA. As for future work, we plan to apply our approach to experimental gene lists and take into account gene-gene relationships (physical Protein Protein Interactions), term-term relationships (Gene Ontology relationships, namely *is-a*, *part-of*, and *regulates*) and relationships between gene positions. Moreover, in order to efficiently deal with the relationships present within the data we can use Relational Concept Analysis.

References

1. Gabriel F. Berriz, Oliver D. King, Barbara Bryant, Chris Sander, and Frederick P. Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–2504, 2003.
2. Scott Doniger, Nathan Salomonis, Kam Dahlquist, Karen Vranizan, Steven Lawlor, and Bruce Conklin. MAPPFinder: using Gene Ontology and GenMAPP to Create a Global Gene-expression Profile from Microarray Data. *Genome Biology*, 4(1):R7, 2003.
3. Aravind Subramanian et al. Gene Set Enrichment Analysis: A Knowledge-based Approach for Interpreting Genome-wide Expression Profiles. *Proceedings of the National Academy of Sciences*, 102:15545–15550, 2005.
4. Arthur Liberzon et al. Molecular Signatures Database (MSigDB) 3.0. *Bioinformatics*, 27(12):1739–1740, 2011.
5. Michael Y. Galperin and Xosé M. Fernández-Suarez. The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Research*, 40(Database-Issue):1–8, 2012.
6. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin/Heidelberg, 1999.
7. Mehdi Kaytoue-Uberall, Sébastien Duplessis, Sergei O. Kuznetsov, and Amedeo Napoli. Two FCA-Based Methods for Mining Gene Expression Data. In Sébastien Ferré and Sebastian Rudolph, editors, *ICFCA*, volume 5548 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2009.
8. Purvesh Khatri and Sorin Draghici. Ontological Analysis of Gene Expression Data: Current Tools, Limitations, and Open Problems. *Bioinformatics*, 21(18):3587–3595, 2005.
9. Sergei O. Kuznetsov. On stability of a Formal Concept. *Ann. Math. Artif. Intell.*, 49(1-4):101–115, 2007.
10. François Rioult, Jean-François Boulicaut, Bruno Crémilleux, and Jérémy Besson. Using Transposition for Pattern Discovery from Microarray Data. In *DMKD*, pages 73–79, 2003.
11. François Rioult, Céline Robardet, Sylvain Blachon, Bruno Crémilleux, Olivier G, and Jean-François Boulicaut. Mining Concepts from Large SAGE Gene Expression Matrices. In *In: Proceedings KDID03 co-located with ECML-PKDD 2003, Catvat-Dubrovnik (Croatia*, pages 107–118, 2003.
12. Camille Roth, Sergei A. Obiedkov, and Derrick G. Kourie. Towards Concise Representation for Taxonomies of Epistemic Communities. In *CLA*.
13. Zhong S, Storch F, Lipan O, Kao MJ, Weitz C, and Wong WH. GoSurfer: a graphical interactive tool for comparative analysis of large gene sets in Gene Ontology space. *Applied Bioinformatics*, 3(4):1–5, 2004.
14. Eran Segal, Nir Friedman, Daphne Koller, and Aviv Regev. A Module Map Showing Conditional Activity of Expression Modules in Cancer. *Nat.Genet.*, 36:1090–8, 2004.

A New Approach to Classification by Means of Jumping Emerging Patterns

Aleksey Buzmakov[1,2], Sergei O. Kuznetsov[2], and Amedeo Napoli[1]

[1] LORIA(CNRS-Inria NGE-Université de Lorraine),Vandoeuvre les Nancy,France
[2] National Research University Higher School of Economics, Moscow, Russia

Abstract. Classification is one of the important fields in data analysis. Concept-based (JSM) hypotheses are a well-known approach to this task. Although the accuracy of this approach is quite good, the coverage is often insufficient. In this paper a new classification approach is presented. The approach is based on the similarity of an object to be classified to the current set of hypotheses: it attributes the new object to the class that minimizes the set of new hypotheses when a new object is added to the training set. The proposed approach provides a better coverage in compare with the classical approach.

Keywords: Classification, Formal Concept Analysis, JSM-Hypotheses, Jumping Emerging Patterns, Experiments

1 Introduction

Data analysis applications play important role in nowadays scientific researches. One of the possible tasks is to predict object properties, for instance, prediction of a molecule toxicity. Objects can be described in different ways, one of them is by a set of binary attributes. For example, in chemistry domain, a molecule could be characterized by a set of functional groups, belonging to the molecule. Given a set of objects, labeled with several classes (like toxic and non toxic), the prediction task is to estimate the class of some unlabeled object.

Jumping emerging patterns (JEP) is a well studied and interesting approach to the classification[1, 2]. Given a set of classes, like toxic or non toxic molecule, a JEP is a set of characteristics describing a class in a unique way (in the same way as a "monothetic" property). For example, a set of functional groups say S is a JEP when all the database molecules, including all functional groups from S, are toxic. Most of the time, JEPs can be ordered, thanks to an ordering relation, and w.r.t. domain knowledge. In particular, this can be found in [3–5] where JEPs are studied through the so-called JSM-hypotheses.

Then, a classical way to classify an object w.r.t JEPs is to search for JEPs, describing the object, and if these JEPs are of the same class say k, then the object should be classified in k. If there is no such JEP or there are JEPs of different classes, the object remains unclassified. Although for the classical approach the prediction accuracy (the probability that the prediction is correct) is quite high, its coverage (the probability that the object attributed to any class by the classifier and this attribution is correct) is rather low. So a new method is proposed with comparable accuracy and much better coverage. The method relies on the MDL (minimal length description) principle, where the outcome class for an object is the class, minimizing the number of associated JEPs.

There are two main objectives in the paper. The first is to connect JEPs with JSM-hypotheses; and the second is to suggest a new classification approach, based on JEPs, and to check it experimentally.

The paper is organized as follows. In Section 2 definitions are introduced. Then Section 3 describes the classical and the new approaches to classification. Section 4 details the computer experiments and their results. And finally, Section 5 concludes the paper.

2 Definitions

2.1 Formal Concept Analysis and Pattern Structures

This section briefly introduces the main definitions on pattern structure in formal concept analysis (see [6]) and emerging patterns (see [1, 2]).

Definition 1. *A pattern structure is a meet-semilattice $(D, \sqcap)$. Elements of a set D are called patterns.*

Definition 2. *A pattern context is a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a pattern structure, and $\delta : G \to D$ is a mapping function from objects to their descriptions.*

The recently studied interval patterns [7] and the pattern structure given by sets of graphs [6] are examples of pattern structures.

Usually a formal context is introduced as follows [8].

Definition 3. *A formal context is a triple (G, M, I), where G is a set of objects, M is a set of attributes and $I \subseteq G \times M$ is a binary relation between G and M.*

A 'classical' formal context (G, M, I) could be considered as a special case of pattern context $(G, (D, \sqcap), \delta)$. The set of objects remains G, $D = 2^M$, with a semilattice operation corresponding to intersection of sets, and $\delta = g \in G \to \{m \in M | (g, m) \in I\}$. For instance a particular context is shown on Table 1. A mapping function δ maps the object g_1 to the set $\{\mathtt{m_1}, \mathtt{m_2}, \mathtt{m_5}, \mathtt{m_6}, \mathtt{m_7}\}$. For the sake of simplicity, all further examples will refer to classical contexts.

Objs\Attrs	m_1	m_2	m_3	m_4	m_5	m_6	m_7
g_1	x	x			x	x	x
g_2	x	x		x		x	x
g_3	x	x			x	x	
g_4		x	x				x
g_5	x			x	x	x	
g_6		x				x	x

Table 1: Formal Context (G, M, I).

Object	Class
g_1	k_1
g_2	k_1
g_3	k_2
g_4	k_2
g_5	k_2
g_6	?

Table 2: Labeling function.

A Galois connection associated to the context $(G, (D, \sqcap), \delta)$ is defined as:

$$A^\diamond = \sqcap_{e \in A} \delta(e), \qquad A \subseteq G \tag{1}$$

$$d^\diamond = \{e \in G | d \sqsubseteq \delta(e)\}, \qquad d \in D \tag{2}$$

For $a, b \in D$, $a \sqsubseteq b \Leftrightarrow a \sqcap b = a$, and the operation $(\cdot)^{\diamond\diamond}$ is a closure operator.

Definition 4. *A pattern $d \in D$ is closed iff $d^{\diamond\diamond} = d$.*

Definition 5. *Generator of a closed pattern $d \in D$ is a pattern $x \in D$, such that $x^{\diamond\diamond} = d$.*

Definition 6. *A pattern concept is a pair (A, d) such that $A \subseteq G$, $d \in D$, $A^{\diamond} = d$, $A = d^{\diamond}$. A is called the extent of the concept and d is called the intent. The intent of a formal concept is a closed pattern (while the extent A is a closed set of objects, i.e. $A^{\diamond\diamond} = A$).*

For example $(\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\})$ is a concept w.r.t the context shown on the Table 1. One of the possible generators of its intent is $\{\mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\}$.

2.2 Classification Concepts

The classification operation can be carried out in FCA using so-called hypotheses. In classification there are a set of classes K and a mapping function $\xi : G \to K \cup \{?\}$, where '?' means unknown class of an object.

Definition 7. *Given a certain class $k \in K$, we note the set of objects belonging to the class k as $G_{k+} = \{g \in G | \xi(g) = k\}$ and the set of objects, which are not belong to class k as $G_{k-} = \{g \in G | \xi(g) \neq k, \xi(g) \neq ?\}$. A hypothesis for class k is a pattern $h \in D$, such that $h^{\diamond} \cap G_{k-} = \emptyset$ and $\exists A \subseteq G_{k+} : A^{\diamond} = h$.*

For example, $\{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\}$ is a hypothesis for class k_1 because $\{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\}^{\diamond} = \{\mathtt{g}_1, \mathtt{g}_2\}$ contains objects of only one class.

In itemset mining Jumping Emerging Patterns (JEP) are used for classification [1, 2]. Although the usual definition of a JEP does not involve pattern structures, it can be convenient to introduce JEP w.r.t pattern structures.

Definition 8. *A pattern $d \in D$ is a JEP for a class $k \in K$ when $d^{\diamond} \neq \emptyset$ and $\forall g \in d^{\diamond}$, $\xi(g) = k$.*

According to definitions 7 and 5, a hypothesis for a class $k \in K$ is a JEP, whereas a JEP for a class $k \in K$ is a generator of some hypothesis for the class k. For the context on Table 1 and ξ function from Table 1 $\{\mathtt{m}_6, \mathtt{m}_7\}$ is a JEP for the class k and it is a generator for $\{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\}$, which is a hypothesis.

3 Classification

This section introduces classification by means of Jumping Emerging Patterns (JEP) in two different ways: the classical approach and the new approach.

For some class $k \in K$, H_{k+} is the set of all JEPs for class k and H_{k-} is the union of JEPs for all other classes. The union of all JEPs is denoted as $H = H_{k+} \cup H_{k-}$.

Definition 9. *A JEP $h \in H_{k+}$ describes an object $g \in G$ if $h \sqsubseteq g^{\diamond}$.*

According to the classical approach [3], a new object g_{new} should be attributed to the class $k \in K$ iff there is a JEP for the class k, describing g_{new} $(\exists h \in H_{k+} : h \sqsubseteq \delta(g_{new}))$, and there is no JEP for other classes, describing the object $(\nexists h \in H_{k-} : h \sqsubseteq \delta(g_{new}))$. This method will be referred as Cl-method.

For example, object g_6 should be attributed to the class k_1 because there exists a JEP for the class k_1, namely $\{\mathtt{m}_6, \mathtt{m}_7\}$, and no JEP for any other class.

In contrast, it is not possible to classify the object with hypotheses, because the corresponding hypothesis would be $\{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_6, \mathtt{m}_7\}$ which does not describe the object g_6.

The classical approach usually works well but there are a lot of objects that may not be classified [9]. Another problem is related to real-world data and interpretation of the classification: one may expect to have only one JEP attributing an object to a class. For instance, in the task of predicting toxicity of a molecule, every JEP is a set of substructures and so ideally it should be the set of those substructures which raises the toxicity of the molecule, while in practice there are a lot of JEPs describing every object and so some of them have no relation to the toxicity-specific set of substructures.

For going in this direction, one could recall a principle, widely used in natural science: among all explanation of phenomena one should select the simplest one. So a set of JEPs in our case should classify as many objects from training set as possible, whereas it should not be too complicated. The whole number of JEPs is rather arbitrary, and so it cannot be a measure of complexity. On the other hand if an object should be attributed to a class by only one JEP, then it is natural to suggest that "important JEPs" a) covers all objects and b) that these JEPs are rather general. So the complexity of a system of JEPs could be measured by the minimal number of JEPs required to describe all the objects attributed to any class.

3.1 Running Example

On Table 3a formal context is shown: real life objects, described by some properties, like color and weight. The objects are labeled whether they are natural or human-made. The given labeling is shown on Table 3b. The task is to predict labels of `Cat` and `Elephant`. Tables 3e-3d are other labeling functions used during classification procedure.

	alive	can move	metal	light	green
Tree	x				x
Fungus	x			x	
Velo		x	x	x	x
Car		x	x		x
Cat	x	x		x	
Elephant	x	x			

(a)

Object	Made by
Tree	Nature
Fungus	Nature
Velo	Human
Car	Human
Cat	'?'
Elephant	'?'

(b)

Obj	M
T	N
F	N
V	H
Car	H
Cat	**N**
El	'?'

(c)

Obj	M
T	N
F	N
V	H
Car	H
Cat	**H**
El	'?'

(d)

Obj	M
T	N
F	N
V	H
Car	H
Cat	'?'
El	**N**

(e)

Obj	M
T	N
F	N
V	H
Car	H
Cat	'?'
El	**H**

(f)

Table 3: Running Example Formal Context. Figures 3b-3f are different correspondences between objects and their classes (ξ-functions).

The JEPs for the context on Table 3a and labeling function on Table 3b are the following: $\mathtt{a}(\text{alive}) \to \mathtt{N}$, $\mathtt{cm}(\text{can move}) \to \mathtt{H}$, $\mathtt{m}(\text{metal}) \to \mathtt{H}$, $\mathtt{l}(\text{light})$, $\mathtt{g}(\text{green}) \to \mathtt{H}$. Neither `Cat` nor `Elephant` may be classified, as they both include JEPs, corresponding to different labels ($\mathtt{a} \to \mathtt{N}$ and $\mathtt{cm} \to \mathtt{H}$). But maybe we are still able to classify them? Let us assume that `Cat` (or `Elephant`) is made by `Nature` (Tables

3c, 3e) and then that they are made by `Human` (Tables 3d, 3f). And then as a response to the classification task we give the class of the best assumption.

Let us assume that the `Cat` is made by `Nature`, the labeling function is shown on Table 3c. The corresponding set of JEPs is as following: `a` $\rightarrow$ `N`; `m` $\rightarrow$ `H`; `l,g` $\rightarrow$ `H`; `cm,g` $\rightarrow$ `H`. We should notice that the label (or class) of every object from Table 3a can be explained by at least one JEP, i.e. for an object g there is a JEP describing object g and corresponding to the class of object g. Let now assume that object `Cat` is made by `Human`, the labeling function is shown on Table 3d. The corresponding set of JEPs is as following: `a,g` $\rightarrow$ `N`; `cm` $\rightarrow$ `H`; `m` $\rightarrow$ `H`; `l,g` $\rightarrow$ `H`. Among these JEPs, there is no JEP explaining the class of object `Fungus`, and so we can say that the assumption that `Cat` is made by `Nature` is better than the assumption that `Cat` is made by `Human`, and so the `Cat` should be classified to class `Nature`.

For the `Elephant` let us assume first that it is made by `Nature`, the labeling function on Table 3e. The set of JEPs are `a` $\rightarrow$ `N`; `m` $\rightarrow$ `H`; `l,g` $\rightarrow$ `H`; `cm,l` $\rightarrow$ `H`; `cm,l` $\rightarrow$ `H`. They explain classes of every object from the context. Let us assume that the `Elephant` is made by `Human`. The set of JEPs are `a,g` $\rightarrow$ `N`; `a,l` $\rightarrow$ `N`; `cm` $\rightarrow$ `H`; `m` $\rightarrow$ `H`; `l,g` $\rightarrow$ `H`. They do also explain all the objects from the context but we are still able to make a good prediction. For that we should calculate the minimal number of JEPs required to explain every object from the set. For the assumption that `Elephant` is made by `Nature`, one requires 2 JEPs to explain every object from the context (`a` $\rightarrow$ `N`; `m` $\rightarrow$ `H`). For the assumption that `Elephant` is made by `Human`, one requires 3 JEPs (`a,g` $\rightarrow$ `N`; `a,l` $\rightarrow$ `N`; `cm` $\rightarrow$ `H`). Thus we could say that although both assumptions are possible, the first one is more simple (require only 2 JEPs for explaining every object from the context) and the `Elephant` should be classified to class `Nature`.

3.2 The New Approach

We have a pattern context $(G, (D, \sqcap), \delta)$ and a set of classes K. Every object in G can either have a class from K or no class, denoted as '?'. A labeling function $\xi : G \rightarrow K \cup \{?\}$ attributes an object g to a class k. Given a context $(G, (D, \sqcap), \delta)$, a set of classes K and a labeling function ξ, one can derive a set of JEPs named H. A system of JEPs refers to a set of all JEPs, derived from a certain context, a certain set of classes, and a certain ξ function.

Definition 10. *A coverage of a system of JEPs H is the set of objects, attributed to some class and described by at least one JEP from H,*
$Coverage(H) = \{g \in G | \xi(g) \neq' ?' \text{ and } \exists h \in H, h \sqsubseteq g^{\diamond}\}$.

Definition 11. *A covering set of JEPs denoted by H^* for a given system of JEPs H is such that:*

- $H^* \subseteq H$;
- *all objects in $Coverage(H)$ are described by at least one JEP from H^*,*
  $\forall g \in Coverage(H) : \exists h^* \in H^* : h^* \sqsubseteq g^{\diamond}$

Definition 12. *For a given system of JEPs H, a size of a minimal covering set of JEPs $MinCover(H)$ is the size of a covering set (for the system) with the minimal number of JEPs among all others covering sets for that system.*

Our approach is based on the above definitions. The definitions consider a JEP only w.r.t. a set of objects described by this JEP. And so any JEP among JEPs describing the same set of objects can be considered, without changing the outcome. It is more efficient to mine only closed patterns. Given a context $(G, (D, \sqcap), \delta)$, one can find a set of concepts and then derive a set of hypotheses H for a given set of classes and a given ξ function. Recall that a hypothesis $d \in D$ is associated to a concept (A, d) and every object in A is labeled by the same class or by '?'. Actually a concept (A, d) will not yield a hypothesis when A includes two objects g_1 and g_2 such that $\xi(g_1) \neq' ?'$, $\xi(g_2) \neq' ?'$ and $\xi(g_1) \neq \xi(g_2)$.

Now we can explain our classification approach. For every unclassified object $g \in G$ the method proceeds as follows:

1. For every class $k_i \in K$, one should change the ξ-function to return class k_i for the object g (instead of '?'), $\xi(g) := k_i$. It leads to changing a system of JEPs to H_i. (For instance, in section 3.1 we assume that `Cat` and `Elephant` are either made by `Nature` or by `Human`).
2. For every system of JEPs H_i one should calculate its coverage ($Coverage(H_i)$). If the assumption $\xi(g) := k_i$ is right, all the objects from $Coverage(H)$ and the object g should be covered by H_i. H_i is called complete if $Coverage(H_i) = Coverage(H) \cup \{g\}$ (In Section 3.1, only the system corresponding to the assumption that `Cat` is made by `Human` was incomplete).
   If there is only one complete system then the corresponding class is considered as a result class (as it was made for `Cat` in Section 3.1).
3. For every complete system H_i one should calculate the size of a minimal covering set of JEPs ($MinCover(H_i)$).
4. The only system minimizing the size of minimal covering set corresponds to the predicting label of the object (In Section 3.1, the assumption that `Elephant` is made by `Nature` brings to 2 JEPs in minimal covering set, and corresponds to the predicted `Elephant` class, i.e. `Nature`). If there are more then one minimizing system then the object is unclassifiable.

The full method will be referred as M1 and the method of only first 2 steps will be referred as M2. In Section 3.1 `Cat` can be classified with M1- and M2-method, contrary the `Elephant` can be classified with only M1-method.

The task of finding minimal cover is NP-complete [10]. It can be shown that difference between minimal covering sets sizes ($|MinCover(H) - MinCover(H_i)|$) of these two systems is often equal to 1. So an approximate solution for the minimal cover set problem can significantly the classification quality.

4 Computer Experiments

Section presents computer experiment and the results.

A database 'Prediction Toxicity Challenge 2000-2001'[3] was used for the experimentation. It consists of molecules labeled by the chemical toxicity with respect to rats and mice of different sexes. Although there are some intermediate labels beside positive and negative. Only positive and negative labels were considered. In Table 4 the sizes of training and test sets are shown.

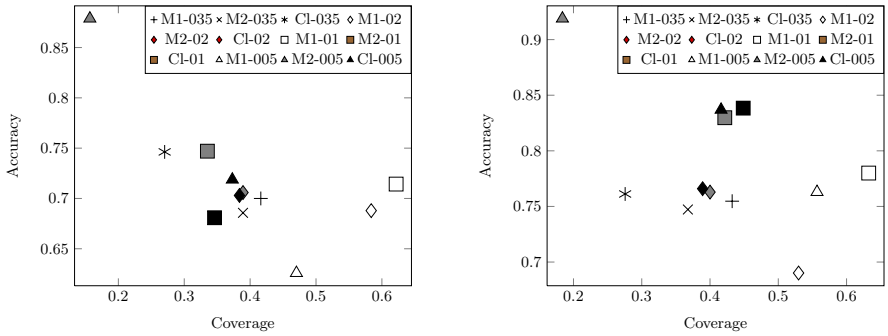
[3] http://www.predictive-toxicology.org/ptc/

	Male Rats	Female Rats	Male Mice	Female Mice
Positives Examples	69	63	68	79
Negatives Examples	192	229	207	206
Test set Positives Examples	84	63	55	66
Test set Negatives Examples	198	219	227	216

Table 4: Numbers of positives and negatives examples in the databases.

One of the way to describe a molecule for applying FCA is to consider it as a graph, where vertices are atoms and edges are bonds between atoms. Then every molecule can be considered as the set of frequent subgraphs, included into the molecule graph. Frequent subgraph means that it is at least present in a certain number of molecules graphs. After converting a set of molecules into graphs, one could use different frequent graph miners [11, 12] to find all frequent subgraphs. Further a frequency limit will be given as percent of the whole molecule set.

To realize M1-classifier one needs a solver for the minimal cover set problem. A greedy algorithm was used to solve the problem approximately. On every iteration the algorithm selects the set, covering the maximal number of uncovered elements. Algorithm stops when all elements are covered by the selected sets.



(a) Male Rats (b) Female Mice

Fig. 1: The Classification Results.

The results for different frequency limit on the database of male rats are shown on Figure 1a and results for female mice are shown on Figure 1b, results for females rats and males mice databases are not shown for the sake of space. Every point on the plots corresponds to the accuracy and coverage of some classifier, while the molecule is considered as a set of frequent substructure. The classifier and the frequency limit are written in the legend.

The quality of M2-classifier is usually higher than the quality of Cl-classifier, whereas coverage of M2-classifier is decreasing with decreasing of frequency limit (increasing the length of description). M2-classifier refers only to the coverage of a system of hypotheses, thus the coverage is an important measure for the classification. The coverage of M1-classifier is much higher then coverage of classical classifier, but the accuracy is worse then for the classical approach, especially in the case of low frequency limit (long description). This could mean that either M1-classifier is over-learned (it became too specific to training set) or it is important for the algorithm to use an exact solution for minimal cover set problem. As it was mentioned in the step 3 of our approach we need to solve a minimum cover set problem, but for the sake of efficiency the greedy algorithm

was used instead of the exact solution. With decreasing of frequency limit the size of minimal cover is increasing, and so an absolute error in defining the size of the minimal cover is increasing as well.

5 Conclusion

In the paper a new approach to classification was suggested. The quality of this approach was checked and it was shown that the number of objects covered by a system of hypothesis is an important characteristic for classification task.

Although the new approach classifies more objects than the classical approach, in some situations it has worse classification quality. One of the possible reasons is an approximate solution for the minimal cover problem. The influence of the approximate minimal cover problem solution should be checked.

References

1. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '99, New York, ACM (1999) 43–52
2. Poezevara, G., Cuissart, B., Crémilleux, B.: Extracting and summarizing the frequent emerging graph patterns from a dataset of graphs. Journal of Intelligent Information Systems **37** (July 2011) 333–353
3. Ganter, B., Kuznetsov, S.: Formalizing hypotheses with concepts. In Ganter, B., Mineau, G., eds.: Conceptual Structures: Logical, Linguistic, and Computational Issues. Volume 1867 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2000) 342–356 10.1007/10722280_24.
4. Blinova, V.G., Dobrynin, D.A., Finn, V.K., Kuznetsov, S.O., Pankratova, E.S.: Toxicology analysis by means of the JSM-method. Bioinformatics **19**(10) (2003) 1201–1207
5. Kuznetsov, S.O., Samokhin, M.V.: Learning closed sets of labeled graphs for chemical applications. In: ILP. (2005) 190–208
6. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: ICCS. (2001) 129–142
7. Kaytoue, M., Duplessis, S., Kuznetsov, S., Napoli, A.: Two FCA-Based methods for mining gene expression data. In Ferré, S., Rudolph, S., eds.: Formal Concept Analysis. Volume 5548 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2009) 251–266 10.1007/978-3-642-01815-2_19.
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997)
9. Helma, C., King, R.D., Kramer, S., Srinivasan, A.: The predictive toxicology challenge 2000–2001. Bioinformatics **17**(1) (2001) 107–108
10. Cormen, T.H.: Introduction to algorithms. MIT Press, Cambridge (2009)
11. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: Proceedings of IEEE International Conference on Data Mining, 2002. (2002) 721 – 724
12. Nijssen, S., Kok, J.: The gaston tool for frequent subgraph mining. Electronic Notes in Theoretical Computer Science **127** (March 2005) 77–87

Semantic querying of data guided by Formal Concept Analysis

Víctor Codocedo[1], Ioanna Lykourentzou[1,2⋆], and Amedeo Napoli[1]

[1] LORIA - CNRS - INRIA - Univesité de Lorraine, BP 239, 54506 Vandœuvre-les-Nancy.
victor.codocedo@loria.fr, amedeo.napoli@loria.fr,
[2] Centre de Recherche Public Henri Tudor - 29, avenue John F. Kennedy L-1855
Luxembourg-Kirchberg, Luxembourg
ioanna.lykourentzou@tudor.lu

Abstract. In this paper we present a novel approach to handle querying over a concept lattice of documents and annotations. We focus on the problem of "non-matching documents", which are those that, despite being semantically relevant to the user query, do not contain the query's elements and hence cannot be retrieved by typical string matching approaches. In order to find these documents, we modify the initial user query using the concept lattice as a guide. We achieve this by identifying in the lattice a formal concept that represents the user query and then by finding potentially relevant concepts, identified as such through the proposed notion of *cousin concepts*. Finally, we use a concept semantic similarity metric to order and present retrieved documents. The main contribution of this paper is the introduction of the notion of *cousin concepts* of a given formal concept followed by a discussion on how this notion is useful for lattice-based information indexing and retrieval.

1 Introduction

As the amount of information grows, the ability to retrieve documents relevant to the needs of the user increasingly becomes more important. Several applications have been proposed, regarding this task, in the field of Information Retrieval (IR). However, as the information becomes more complex (not only text, but also multimedia documents) and specific (domain-oriented), the capacity to organize it becomes as important as the capacity to retrieve it.

Formal Concept Analysis (FCA) is a robust and widely used framework to organize objects based on their relations through their attributes in a concept lattice [6]. Concept lattices have been used in the past to support Information Retrieval tasks and they have been found to have better or comparable performance in relation to traditional approaches, such as Hierarchical Clustering and Best-Match Ranking. We argue that this performance can be further enhanced considering features as concept and semantic similarities and lattice navigation techniques.

⋆ The work of Ioanna Lykourentzou in the present project is supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND).

In this work, we present an approach to retrieve documents from a document-term concept lattice, considering that concepts can be *close* with respect to their position within the lattice and semantically *similar* to one another. We use both of these notions to find which are the most relevant documents for a given user query.

The rest of this paper is organized as follows: Section 2 presents the related research literature. Section 3 briefly introduces FCA and presents our proposed approach for navigating the lattice using the notion of *cousin concepts*, as well as for ranking the selected concepts with respect to their semantic similarity. Section 4 presents and discusses the experimental results and finally section 5 presents the conclusions of our work.

2 Related Work

2.1 Concept lattice-based Information Retrieval

Formal concept analysis is a data representation, organization and management technique with applications in many fields of information science, ranging from knowledge representation and discovery to logic and AI [15]. In the last decade researchers have also focused on examining the potential of FCA addressing problems in the field of Information Retrieval [14]. Under this light, the term Concept lattice-based Information Retrieval is used to describe the problem of retrieving information relevant to a given user query, when the document collection that contains the information is organized within a concept lattice. Some of the IR tasks that FCA and concept lattices have so far been applied on, include query refinement and expansion, integration of query and navigation and support of faceted search ([4, 2]). Among the most representative works in the field are the works of Carpineto and Romano, who introduce the method of Concept lattice-based ranking (CLR) [1].

The CLR method consists of three main steps: i) construction of the formal context of documents-terms and building of the corresponding concept lattice ii) insertion in the lattice of a new concept that represents the user query, using a subset of the attributes of the formal context and iii) retrieval and ranking of the relevant concepts using a nearest-neighbour approach, which depends on their topological path distance, within the lattice, from the original concept. The topological path metric used is called distance ”ring”, and it measures the radius of distance between two concepts, using as distance metric the length of the shortest path between them. The ring metric provides a partially ordered retrieval output, according to which all the documents that are equally distant from the original concept, i.e. belong to the same distance ring, are given the same ranking score.

Carpineto and Romano, also compare the CLR method with two other Information retrieval methods, namely Hierarchical Clustering-based Ranking (HCR)[7] and Best-match ranking (BMR). CLR is found to produce better results compared to HCR. Compared to BMR, it produces worst results when compared on the retrieval over the total document collection and better results, when only the first documents of the retrieval result are considered. However, CLR was better over both BMR, HCR when considering the retrieval of non-matching documents, (i.e. documents that do not match

the user query but share common terms with documents that do match the user query) The main advantage of the CLR method is that, in contrast to other statistical similarity measures that calculate the distance between two document representations using only the characteristics of those representations, the lattice allows to also incorporate the similarity that two document representations have in regards to the context, i.e. the whole document of collections, in which they are found.

The limitations of the traditional CLR method include firstly, the need to build the whole lattice before retrieving the related concepts. This issue determines the complexity and computational time required to address the problem, and it may result in non-realistic solutions for large document collections, like for instance the TREC dataset ([4]). Another issue, identified by the authors is that perhaps CLR should be combined with BMR, since they perform well in different types of documents (non-matching and matching respectively). Another set of limitations, more related to the present work, refers to the ranking method used and specifically to the fact that the retrieval and ranking of the related concepts is made taking into account only their topological relation with the original user query concept. Specifically, due to the use of topological distance rings as a metric of concept similarity, the CLR method does not distinguish between generalization and particularization, when moving from the concept of the original user query to other concepts. This limitation is critical, as it may lead to a loss of the semantic similarity between the retrieved and the original concept and it is explained in more detail in the section 3 when introducing our proposed method for concept-based information indexing and ranking.

To address these limitations, in this paper we propose a novel approach, which seeks to ensure semantic similarity with the original user query, both through the way that the lattice is traversed and through the way that the concepts are ranked. In particular, we introduce a new topological-based concept characteristic, called *cousin concepts*, to navigate the lattice and retrieve candidate related concepts. In parallel, for ranking the retrieved concepts we do not rely only on structural concept similarity features, but instead we use a metric that allows the weighting of both structural and semantic similarity aspects [5].

3 Methodology

In order to present our approach, first we present a brief description to Formal Concept Analysis (FCA). The basics of FCA are introduced in [6], but we recall some notions useful for its understanding in the following.

Data is encoded in a formal context $\mathcal{K} = (G, M, I)$, i.e. a binary table where G is a set of objects, M a set of attributes, and $I \subseteq G \times M$ an incidence relation. Two derivation operators, both denoted by $'$, formalize the sharing of attributes for objects, and, in a dual way, the sharing of objects for attributes:

$' : \wp(G) \longrightarrow \wp(M)$ with $A' = \{m \in M \mid \forall g \in A, gIm\}$

$' : \wp(M) \longrightarrow \wp(G)$ with $B' = \{g \in G \mid \forall m \in B, gIm\}$,

where $\wp(G)$ and $\wp(M)$ respectively denote the powersets of G and M. The two derivation operators $'$ form a *Galois connection* between $\wp(G)$ and $\wp(M)$. The maximal sets of objects which are related to the maximal sets of attributes correspond to closed

sets of the composition of both operators $'$, denoted $''$, for $\wp(G)$ and $\wp(M)$ respectively. A pair $(A, B) \in \wp(G) \times \wp(M)$, where $A = B'$ and $B = A'$, is a *formal concept*, A being the *extent* and B being the *intent* of the concept. The set $\mathcal{C}_\mathcal{K}$ of all concepts from $\mathcal{K}$ is ordered by extent inclusion, denoted by $\leq_\mathcal{K}$, i.e. $(A_1, B_1) \leq_\mathcal{K} (A_2, B_2)$ when $A_1 \subseteq A_2$ (or dually $B_2 \subseteq B_1$). Then, $\mathcal{L}_\mathcal{K} = \langle \mathcal{C}_\mathcal{K}, \leq_\mathcal{K} \rangle$ forms the *concept lattice* of $\mathcal{K}$.

Typically, a concept lattice to index documents is created from a formal context $\mathcal{K}_{index} = (G, M, I)$ where G is a set of documents and M is a set of terms. Thus, the set I represents document *annotations* (i.e. gIm indicates that the document g is annotated with the term m). In a nutshell, to retrieve documents given a conjunctive query[3] $q = \{m_i\}, m_i \in M$ (m_i in the query are hereafter referred as *keywords*), the goal is to find those formal concepts (A, B) where $B \sim q$ and to retrieve the documents in A. The usual approach is to insert into the lattice a *query concept* $C_q = (\{\emptyset\}, q)$ [11, 10, 1] the intent of which contains all the keywords in the user query. Different techniques have been proposed to navigate the lattice, however they rely on topological properties (navigating the super-concepts and sub-concepts if C_q) of the concept lattice to search for documents. Although topology-based measures are useful to retrieve *related documents* from a query, there are some drawbacks that could be overcome with the use of semantic similarity.

The first disadvantage with the navigating in the hierarchy of C_q refers to the generalization of the query. By obtaining the super-concepts of the *query concept* inserted in the lattice, a level of granularity already provided by the user is lost. For example, for a query of the form *"complications, arthroscopy"*, a query concept $C_q = (A_q = \{\emptyset\}, B_q = \{complications, arthroscopy\})$ is created within the lattice. Any super-concept $C_{sup} = (A_{sup}, B_{sup})$ of the query concept has to comply with $B_{sup} \subset B_q$. In this case, only three super-concepts can be obtained: $C_{sup1} = (A_{sup1}, \{$*complications*$\})$, $C_{sup2} = (A_{sup2}, \{$*arthroscopy*$\})$ and $C_{sup3} = (A_{sup3}, \{\emptyset\})$. However, A_{sup1} contains documents about *complications* in any aspect leading to a decrease in precision. The same happens with documents in A_{sup2} containing documents about *arthroscopy* in general whether the user had already specified a restriction for them. C_{sup3} represents the supremum where A_{sup3} contains every possible document, this, of course, is the worst case scenario where the system has no restrictions to retrieve documents.

The second disadvantage is about the specification of the query. By obtaining the whole set of sub-concepts of the query concept the system assumes restrictions not provided by the user. While this is the main idea behind *query expansion* [3] the problem is that there are no discrimination with the sub-concepts that should be used to retrieve documents. For example, given the same query used in the last example and the sub-concepts $C_{sub1} = (A_{sub1}, \{complications, arthroscopy, infection\})$ and $C_{sub2} = (A_{sub2}, \{complications, arthroscopy, practice\})$, the system cannot decide whether the documents in A_{sub1} or the documents in A_{sub2} are the most relevant. From a human perspective, it could be assumed that documents in A_{sub1} may be of more interest for the user since an *infection* is a possible *complication* in the context of a surgery such as an *arthroscopy* and hence they should be retrieved first. On the other side, *practice* is a general word which may lead to non-relevant documents.

[3] Keywords and the conjunction operator $\wedge$

Regarding these problems we propose a technique to improve information retrieval based on concept lattices using the idea of "concept similarity" provided by Formica. We combine this idea with a novel heuristic to navigate the lattice in order to find those concepts holding relevant documents for a given query.

3.1 Navigating the lattice

Given the formal context $\mathcal{K}_{index}$ and the query $q = m_i$ at the beginning of this section, a very simplistic approach to retrieve documents relevant to the query is to find those concepts (A_j, B_j) where $m_i \in B_j : \forall m_i \in q$ defined in [13] as *retrieve algorithm*. Actually, it is possible to find a single concept $C_q = (A_q, B_q)$, where B_q holds the minimal set of words containing all keywords. Subsequently, A_q contains the maximal set of documents containing all the keywords. We refer to $C_q = (A_q, B_q)$ as the matching concept.

It should be noted here that, for a given query q, the matching concept C_q may not exist. This is more likely to happen if the number of keywords is high. In a complete concept lattice (not filtered through any means and constructed using the total amount of information), such a case would actually mean that there are no documents which comply with all the restrictions provided in the user's query. While some strategies can be implemented to overcome this issue (asking the user to provide a simpler query or manipulating the query in order to answer) for the scope of this work we do not elaborate on this and we rather consider the case of an existing matching concept.

Once the matching concept $C_q = (A_q, B_q)$ is found, all documents in A_q can be retrieved to the user. Since the number of documents in A_q may be not sufficient, what is important, in the following, is how to complete the answer with more documents using the lattice.

A simple strategy would involve the hierarchy of C_q, however every sub-concept $(A_{sq}, B_{sq}) \leq_{\mathcal{K}} (A_q, B_q)$ will provide no different documents than those in A_q since $A_{sq} \subset A_q$. Super-concepts of C_q are not useful either because of the problems described in the beginning of this section regarding generalization. Hence, in order to complete the answer with more documents, it is necessary to obtain from the concept lattice some formal concepts which are neither super- nor sub-concepts of (A_q, B_q). To achieve this, we use the notion of *cousin concepts* defined as follows.

Definition of cousin concepts: Two concepts (A_1, B_1) and (A_2, B_2) which are not comparable for $\leq_{\mathcal{K}}$ are said to be *cousins* iff there exists $(A_3, B_3) \neq \perp$ such that $(A_3, B_3) \leq_{\mathcal{K}} (A_1, B_1)$ and $(A_3, B_3) \leq_{\mathcal{K}} (A_2, B_2)$ and $d_{\mathcal{K}}((A_2, B_2), (A_3, B_3)) = 1$ (where $\perp$ is the bottom concept and $d_{\mathcal{K}}$ measures the minimal distance between two formal concepts in the lattice $\mathcal{K}$). Intuitively, this means that (A_1, B_1) and (A_2, B_2) do not subsume each other and that (A_3, B_3) can be either the lower bound or be subsumed by the lower bound $(A_1, B_1) \sqcap (A_2, B_2)$ (where $(A_1, B_1) \sqcap (A_2, B_2)$ denotes the lower bound of (A_1, B_1) and (A_2, B_2).

The use of cousin concepts allows us to move in the lattice from one concept to another using the relations that the elements in their intents possess and that are expressed through their common subsumer. In the example on Figure 1, C_2 is a cousin concept of C_1 because of concept C_3. The attributes *"arthroscopy"*, *"complication"* and *"infection"* are all related through the intent of concept C_3. In this small example, if C_1 is

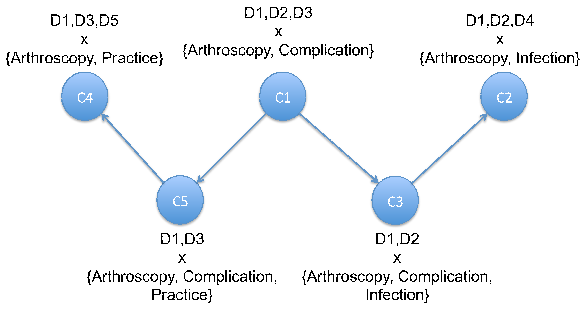


Fig. 1. Example. Five concepts within a lattice, extents and intents are shown. Arrows indicate query *expansion* and *modification*

the matching concept, moving from it to concept C_2 is the same as replacing the word *"complication"* with the word *"infection"* in the query. The extent of concept C_2 will contain documents, some of which are different from those of C_1 and therefore they can be used to complete the answer provided by C_1.

We may also notice that the use of C_3 works as a *query expansion*, adding attributes to the original user query, while the use of C_2 works as a *query modification*, since its attributes are a subset of the attributes of C_3.

Using the entire sub-hierarchy of the matching concept (excluding the infimum) allows us to retrieve several cousin concepts which can be used to complete the answer far beyond the initial set of documents contained in the matching concept's extent. Each cousin concept is a possible *query modification* obtained from a *query expansion*, represented by the sub-concepts of the matching concept.

Although cousin concepts are useful to expand the answer by representing query modifications, their use may entail the same problem described in the beginning of this section, as the second disadvantage of structure-based concept retrieval. In the same example on Figure 1 concepts C_2 and C_4 are cousin concepts of C_1. However, in this scenario the system cannot decide which set of documents, between those of C_2 and C_4, should be retrieved first.

A way to rank cousin concepts is therefore necessary in order to decide which documents should be retrieved to the user first. In this paper we do so using the measure of concept similarity proposed by Formica [5].

3.2 Concept ranking through similarity

The ranking of the retrieved cousin concepts is performed using a semantic similarity metric proposed by Formica[5]. That is, given two formal concepts $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$ the similarity between them is defined as:

$$sim(C_1, C_2) = \frac{|A_1 \cap A_2|}{max(|A_1|, |A_2|)} * w + \frac{\mathcal{M}(B_1, B_2)}{max(|B_1|, |B_2|)} * (1 - w) \qquad (1)$$

where $0 \leq w \leq 1$ is a weighting parameter and $\mathcal{M}(B_1, B_2)$ is the maximization of the sum of the *information content* similarities between each possible pair of terms created using one term from B_1 and another from B_2. *Information content* similarity between two terms is measured using their distance in a lexical hierarchy and/or their co-occurrence in a text corpus. The full explanation of this metric is beyond the scope of this paper. For further information, the reader is referred to the original work of Formica [5].

Consider the example of Figure 1: Concepts C_2 and C_4 are both cousin concepts of the matching concept C_1, and they have the exact same structural features, i.e. the cardinalities of the intersections of their extents/intents with the matching concept are the same, as well as their extent/intent cardinalities. However, when using the semantic similarity metric defined above with $w = 0.5$ and Wordnet[4] as the external lexical hierarchy, we observe that $sim(C_1, C_2) = 0.7275$, while $sim(C_1, C_4) = 0.45$, because the pair *(complication, infection)* has a higher semantic relation than the pair *(complication, practice)*. In this way, we may rank and retrieve the documents of concept C_2, higher than those of concept C_4. Differentiations in the weight value w allow for differentiations in the preference over the structural (from the extents) and semantic (from the intents) similarities of the compared concepts.

4 Experimental results and discussion

We applied our approach using the MuchMore[5] dataset, which contains annotated medical document abstracts (7822 documents, 9485 single or multi-word terms). In order to answer a given user query we follow a 3-step knowledge discovery process, as follows.

Step 1 - Data preprocessing: Pre-filter the set of documents and terms Since the creation of a lattice containing the full set of documents/terms would be computationally expensive, we create a reduced lattice for each given user query. To do so, we implement a simple pre-filtering strategy of iterative expansion similar to the one described in [4]. Given a conjunctive query $q = \{t_i\}$ we fetch all documents d_n that contain all the keywords in the query. Afterwards, we obtain all the terms t_j that these documents contain. Finally, we fetch all the additional documents d_m which contain any of these terms. At the end we obtain a set of $d_n + d_m$ documents and $t_i + t_j$ terms which is used to create a formal context. For the query $q_s = \{$*"complication","arthroscopy"*$\}$, this process returns a set of 11 initial documents, which in turn leads to an expanded set of 177 terms and 7560 documents.

Unfortunately, this strategy yields more than 95% of the corpus' documents because of highly frequent terms. To avoid this, documents with a number of terms below the average (in the above example, 7 terms) are not included in the expanded set of documents (3485 documents for the example). It should be noted that the pre-filtering strategy can be further improved considering weighting techniques such as tf.idf, pivoted normalized document length [9] or heuristic approaches specifically focusing on the reduction of irrelevant concepts in a FCA lattice [4].

[4] Wordnet is a widely-used free semantic dictionary organized in a hierarchical manner [12]

[5] `http://muchmore.dfki.de/`

Step 2 - Transformation: Concept lattice creation The creation of the concept lattice is straightforward since we rely on a fixed framework (Coron Toolkit[6]). For the example of query q_s we obtain 134718 formal concepts without using support pruning.

Step 3 - Data mining & Evaluation: Retrieving documents from the lattice The retrieval step consists of three sub-steps, described in the following.

1. **Find the matching concept.** We search for the matching concept C_q in the lattice using a level-wise algorithm, starting from the supremum. The matching concept C_q is the closer concept to the supremum which contains in its intent all the keywords provided in the query. The existence of the matching concept is predicated in the assumption of a conjunctive query to pre-filter the dataset and create the formal context. In the case that there are no documents containing at least all the keywords, the query is consider unsuccessful and the retrieval process is stopped at step 1. The documents in the extent of the matching concept are retrieved to the user and they are hereafter referred to as *exact answer*.
2. **Find the cousin concepts of the matching concept.** Cousin concepts are obtained for the matching concept and for each of its sub-concepts C_i. A list called *candidate answers* is created storing the pair (C_i, C_j) where $C_i \leq C_q$ and C_j is a cousin concept of C_i. For q_s, the *candidate answers* list contains 2301 (concept, cousin concepts) pairs.
3. **Rank the cousin concepts.** The ranking process is performed using the similarity measure described in section 3.2. Every pair (concept, cousin concept) from the *candidate answers* list is compared, or what is the same, each *query expansion* is compared to its correspondent *query modification*. Formica's concept similarity was implemented using Wordnet [12] as a lexical hierarchy[7], the *Brown corpus* as a base to obtain term frequencies and a modified version of the Hungarian algorithm [8] to match terms from both intents[8]. The experiments here presented where performed with a value of $w = 0.5$.

Table 2 shows the results for two queries executed using the described approach. For $q_s = \{$*"arthroscopy"*,*"complication"*$\}$ the *exact answer* retrieved 11 documents of which 7 are relevant to the user. The *close answer*, composed of the documents retrieved from the ranked cousin concepts, contains 100 documents of which 6 are relevant to the user. Therefore, out of the 21 documents relevant to the user the approach was able to retrieve 13.

It is of special interest to analyse the characteristics of the obtained results. The cousin concept with intent *joints, surgical aspects, complication, diagnostic* has a similarity of 0.71 with the concept with intent *arthroscopy, surgical aspects, complication, diagnostic* which is a sub-concept of the matching concept created for q_s and hence, does not have additional documents than those already retrieved. What can be appreciated here is that the algorithm works firstly by expanding the original query with related

[6] `http://coron.loria.fr/site/index.php`

[7] Wordnet is a dictionary where terms are grouped by synonymia (synset) and ordered in a hierarchical tree by the hypernym relation.

[8] The Hungarian algorithm minimizes the sum of values in the diagonal of a square matrix.

terms (from *arthroscopy* to *surgical aspects*[9]) and secondly by modifying the expanded query with a semantically similar term (from *arthroscopy* to *joints*). The above process is illustrated in Table 1.

Table 1. Query expansion and modification.

matching concept		sub-concept				cousin concept
query		*expansion*				*modification*
arthroscopy	→	arthroscopy	→	~~arthroscopy~~	→	joints
complication		complication		complication		complication
		surgical aspects		surgical aspects		surgical aspects
		diagnostic		diagnostic		diagnostic

The second query in Table 2 is also of interest in the sense that it indicates algorithm robustness. The word laparoscopic is not present in Wordnet, making it not suitable for the comparison in the similarity measure. This means that *laparoscopic* can be replaced with any other term since the algorithm is not able to measure the difference. However, since the similarity measure relies also in extent intersection, the algorithm will try to replace *laparoscopic* with terms used by documents similar to those in the exact answer. In that way, the first ranked close answer is correct and its intent is *complication, risk, cholecystectomy*. Notice that in this case the algorithm does not conclude that the term *risk* is semantically close to the term *laparoscopic*, but that it is the best term to replace the latter in the query.

Table 2. Results for two queries.

Query	**Exact answer** correct/found	**Close Answer** correct/found	**Total Answers** correct/expected
arthroscopy, complication	7/11	6/100	13/21
complication, laparoscopic cholecystectomy	3/3	3/100	6/7

5 Conclusions

In this paper we present a technique to use a concept lattice for the retrieval of documents from a given user query. The proposed technique differs from previous approaches in two main aspects: the lattice navigation algorithm is not restricted to the

[9] an arthroscopy is a knee surgery

hierarchy of the query concept and the ranking algorithm is based on the semantic similarity, rather than on the structural characteristics of the compared concepts.

In terms of navigation, we introduce the notion of *cousin concepts*, which represents *query modifications* that can be used to retrieve documents different from those directly related to the query. In terms of ranking, we use external knowledge sources (a lexical hierarchy and a text corpus) to measure semantic similarity and order the retrieved *cousin concepts* by relevance to the initial query.

We illustrate our approach using two examples from a dataset of medical document abstracts. We also explain certain limitations of the proposed approach, mainly regarding the performance the concept lattice construction and the availability of the terms of the user query in the dataset. Currently, we are applying this approach on the same dataset but on a full scale, in order to measure precision and recall, as well as to compare the proposed technique with other Information Retrieval state-of-the-art techniques.

References

1. Claudio Carpineto and Giovanni Romano. Order-theoretical ranking. *Journal of the American Society for Information Sciencies*, 51(7):587–601, May 2000.
2. Claudio Carpineto and Giovanni Romano. Using concept lattices for text retrieval and mining. In *Formal Concept Analysis*, pages 161–179. Springer-Verlag, Berlin, Heidelberg, 2005.
3. Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1), January 2012.
4. Claudio Carpineto, Giovanni Romano, and Fondazione Ugo Bordoni. Exploiting the potential of concept lattices for information retrieval with credo. *Journal of Universal Computer Science*, 10:985–1013, 2004.
5. Anna Formica. Concept similarity in formal concept analysis: An information content approach. *Knowledge-Based Systems*, 21(1):80 – 87, 2008.
6. Benrhard Ganter and Rudoph Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.
7. Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of SIGIR 1996*, SIGIR '96, pages 76–84. ACM, 1996.
8. H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, pages 83–97, 1955.
9. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
10. Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone. Using domain knowledge to guide lattice-based complex data exploration. In *Proceedings of the 2010 conference on ECAI 2010*, pages 847–852. IOS Press, 2010.
11. Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smal-Tabbone. Querying a bioinformatic data sources registry with concept lattices. In *Proceedings of ICCS 2005*, pages 323–336. LNCS 3596, Springer, 2005.
12. George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.
13. Ibtissem Nafkha, Samir Elloumi, and Ali Jaoua. Using concept formal analysis for cooperative information retrieval. In *Concept Lattices and their Applications 2004*, volume 110 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.
14. Uta Priss. Lattice-based information retrieval. *Knowledge Organization*, 27:132–142, 2000.
15. Uta Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology.*, 40(1):521–543, December 2006.

Information retrieval by on-line navigation in the latticial space-search of a database, with limited objects access

Ch. Demko♦★, K. Bertet♦

♦ L3I - Université de La Rochelle - av Michel Crépeau - 17042 La Rochelle
cdemko,kbertet@univ-lr.fr
★ Joomla! Production Leadership Team
christophe.demko@joomla.org

Abstract. We propose in this paper basic operations with limited access to the objects in the table, which can improve the computation time. Experiments were conducted with Joomla!, a content management system based on relational algebra, and located on a MySQL database. This work follows the results presented in [5].

keywords: concept lattice ; databases ; algorithm ; closure operator

1 Introduction

Galois lattice is a graph providing a representation of all the possible correspondences between a set of *objects* (or examples) O and a set of binary *attributes* (or features) I. *Galois lattices* (or *concept lattices*) were first introduced in a formal way in the graph and ordered structures theory [2,1,4].

The concept lattice is a rich and flexible navigation structure automatically derived from the context, and can therefore be considered as a dynamic and complete space search enables data description while preserving its diversity. Querying and navigation can be freely combined: to each user request corresponds a concept of the lattice as answer ; the user can then improve its search either by amending its request, or by on-line browsing arround the concept in the lattice structure.

In this paper, we propose an implementation of these basic operations with Limited Object Access, aiming to improve time computation for a large amount of objects in large databases.

This paper is organized as follows. In section 2, we describe the concept lattice and the closed set lattice. In section 3, we present the motivations that have conducted this word. In section 4, we describe our basic operations with limited object access. In section 5, we present some experiments.

2 Concept lattice: definition and generation

Definition. The *concept lattice* is a particular graph defined and generated from a a *binary table* (also denoted a *formal context*) C described by a relation R between a set of objects O and a set of attributes I. We associate to a set of objects $A \subseteq O$ the set $f(A)$ of attributes in relation R with the objects of A:

$$f(A) = \{y \in I \mid xRy \ \forall \ x \in A\}$$

Dually, to a set of attributes $B \subseteq I$, we define the set $g(B)$ of objects in relation with the attributes of B:

$$g(B) = \{x \in O \mid xRy \ \forall \ y \in B\}$$

These two functions f and g defined between objects and attributes form a *Galois correspondence*. A *formal concept* represents maximal objects-attributes correspondences (following relation R) by a pair (A, B) with $A \subseteq O$ and $B \subseteq I$, which verifies $f(A) = B$ and $g(B) = A$. The whole set of formal concepts thus corresponds to all the possible maximal correspondences between a set of objects O and a set of attributes I. Two formal concepts (A_1, B_1) and (A_2, B_2) are in relation when they verify the following inclusion property:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow \left\| \begin{array}{l} A_2 \subseteq A_1 \\ \text{(equivalent to } B_1 \subseteq B_2\text{)} \end{array} \right.$$

3 Motivations

The existence of a concept lattice underlying a data table allows to consider an information retrieval strategy combining querying and navigation by a browsing in this lattice as in an area of research. Indeed, the user request is a concept of a lattice, and the user can then improve its search either by amending its request, or by browsing in the lattice structure. From a computational point of view, such a mechanism of information retrieval by request and by navigation requires two main operations:

1. Generation of the smallest concept $(g(B), f(g(B))$ containing a given subset B of attributes: B is the request, the objects part $f(g(B))$ of the concept is the answer, $g(B)$ are inferred attributes.
2. Generation of the immediates successors of a given concept (A, B) for a browsing in the lattice by computing the inclusion-maximal in the set system $\mathcal{F}_A$ defined on O by $\mathcal{F}_A = \{g(x + B) \ : \ x \in I \backslash B\}$.

Large data are often described by a huge amount of objects, as in databases for example where the number of recordings (i.e. objects) can be huge, indexed using sophisticated key-indexation techniques. We propose in this paper an improvement of these basic operations with two limited objects access strategies:

A storage improvment by considering the restriction of the concept lattice to the attributes, namely the *closed set lattice*.

A computation improvment by considering in the operations the cardinality of the subset of objects instead of the subset itself, using the *count function*.

Name: `Immediates_Successors_LOA`
Data: A context K ; A closed set B of the closed set lattice $(\mathbb{C}_I, \subseteq)$ of K
Result: The immediate successors of B in the lattice
begin
  initialize the $Succ_B$ family to an empty set;
  foreach $x \in I \setminus B$ **do**
    add = true;
    foreach $X \in Succ_B$ **do**
      \\ **Merge x and X** in the same potential successor
      if $c(B + x) = c(B + X)$ **then**
        if $c(B + X + x) = c(B + x)$ **then**
          replace X by $X + x$ in $Succ_B$; add=false; break;
        end
      end
      \\ **Eliminate x** as potential successor
      if $c(B + x) < c(B + X)$ **then**
        if $c(B + X + x) = c(B + x)$ **then** add=false; break;
      end
      \\ **Eliminate X** as potential successor
      if $c(B + x) > c(B + X)$ **then**
        if $c(B + X + x) = c(B + X)$ **then** delete X from $Succ_B$
      end
    end
    \\ **Insert x** as a new potential successor ;
    if *add* **then** add $\{x\}$ to $Succ_B$
  end
  return $Succ_B$;
end

Algorithm 1: Generation of the immediate successors of a closed set in the Hasse diagram of the lattice $(\mathbb{C}_I, \subseteq)$

Closed set lattice. Instead of a concept lattice, it is possible to consider its restriction to the attributes in order to limit the storage of huge ammount of objects in each concept. A nice result establishes that any concept lattice $(\mathbb{C}, \leq_C)$ is isomorphic to the lattice $(\mathbb{C}_I, \subseteq)$ defined on the set I of attributes, with $\mathbb{C}_I$ the restriction of $\mathbb{C}$ to the attributes in each concept. The lattice $(\mathbb{C}_I, \subseteq)$ is also known as the closed sets lattice on the attributes I of a context (O, I, R), where the set system $\mathbb{C}_I$ is composed of all closed set - i.e. fixed points - for the closure

operator $\varphi = g \circ f$ - i.e. a map that is isotone, extensive and idempotent):

$$\mathbb{C}_I = \{\varphi(X) \ : \ X \subseteq I\} \tag{1}$$

The closed sets $\bot = \varphi(\emptyset) = f(O)$ and $\top = I$ respectively correspond to the bottom and the top of the closed set lattice. See the survey of Caspard and Monjardet [3] for more details about closed set lattices.

Therefore, each smallest concept $(g(B), f(g(B))$ containing a given set B of attributes is replaced by the closure $\varphi(B) = f(g(B)$ on the attributes. Thus, only the attributes part is stored.

The count function. Moreover, we propose to reinforce the object access limitation by considering the cardinality of the subset $g(B)$ instead of the subset itself in the treatment. The count function c associates to any subset X of attributes the cardinality of the subset $g(X)$: $c(X) = |g(X)|$

It corresponds to the notion of support introducing in rules extraction from data-bases, and is in particular used by Titanic algorithm [6]. We use the count function c instead of the closure operator φ since c and φ possesses together nice properties, $\forall X, X' \subseteq I$:

$$X \subseteq Y \Rightarrow c(X) \geq c(Y) \tag{2}$$

$$\varphi(X) = \varphi(Y) \Rightarrow c(X) = c(Y) \tag{3}$$

$$X \subseteq Y \text{ and } c(X) = c(Y) \Rightarrow \varphi(X) = \varphi(Y) \tag{4}$$

4 Basic operations in a lattice with limited objects access

Using the closed sets lattice $(\mathbb{C}_I, \subseteq)$ instead of the whole concept lattice $(\mathbb{C}, \leq_C)$ gives raise to a storage improvement since only the attributes part is stored. Using the count function, the two main operations can therefore be reformulated as follows:

Generation of the closed-set $\varphi(B)$ of a subset B of attributes. It can be performed using the following equality:

$$\varphi(B) = B + \{x \in I \backslash B \ : \ c(B) = c(B + x)\} \tag{5}$$

This equality is a direct consequence of the third property of c together with the isotone property of φ (i.e. $X \subseteq X' \Rightarrow \varphi(X) \subseteq \varphi(X')$).

Generation of the immediates successors of a closed set B. A closed sets lattice can be generated using an algorithm similar to Bordat's algorithm, where the immediate successors of a closed set B in the Hasse diagram of the lattice are the inclusion-minimal subsets of

$$\mathcal{F}_B = \{\varphi(B + x) \ : \ x \in I \backslash B\} \tag{6}$$

In our previous work [5], we present an incremental algorithm by testing, for each attribute x of $I \backslash B$ and each already inserted potential successor $X \subseteq I \backslash B$, the inclusion between $\varphi(B + X)$ and $\varphi(B + x)$:

1. Merge x with X when $\varphi(B + x) = \varphi(B + X)$.
2. Eliminate X as potential successor of B when $\varphi(B + x) \subset \varphi(B + X)$
3. Eliminate x as potential successor of B when $\varphi(B + X) \subset \varphi(B + x)$
4. Insert x as potential successor of B when x is neither eliminated or merged with X.

The inclusion test between $\varphi(B + X)$ and $\varphi(B + x)$ can easily be performed using the count function c and the following proposition deduced from Prop. 1 in [5]:

Proposition 1. $\varphi(B + X) \subseteq \varphi(B + x) \iff c(B + X + x) = c(B + X)$

$\Rightarrow$: Consider that $\varphi(B + X) \subseteq \varphi(B + x)$. The equivalence between inclusion and intersection set operations ($C \subseteq D \iff C = C \cap D$) allows to deduce that $\varphi(B + X) = \varphi(B + X) \cap \varphi(B + x)$. Since $\varphi(B + X) \cap \varphi(B + x) = \varphi(B + X) \wedge \varphi(B + x) = \varphi(B + X + x)$, then $\varphi(B + X) = \varphi(B + X + x)$. We conclude by $c(B + X + x) = c(B + X)$ using the second property of the count function c (see Eq. 3).
$\Leftarrow$: Consider that $c(B + X + x) = c(B + X)$. By Eq. 4, and since $B + X \subseteq B + X + x$, we deduce that $\varphi(B + X + x) = \varphi(B + X)$, and we conclude by $\varphi(B + X) \subseteq \varphi(B + x)$ as above.

In the case where $\varphi(B + X) \subseteq \varphi(B + x)$, the strict inclusion has then to be tested in order to decide if x has to be deleted as potential successor, or merged with X. Using Eq. 2 and Eq. 3, this test can be performed by checking if $c(B+X) > c(B+x)$ or $c(B+X) = c(B+x)$. The case where $\varphi(B+x) \subseteq \varphi(B+X)$ is dualy tested in order to decide if X has to be deleted or not as potential successor.

The complexity of computing the immediate successors of a closed set B using the `Immediates_Successors_LOA` algorithm is:

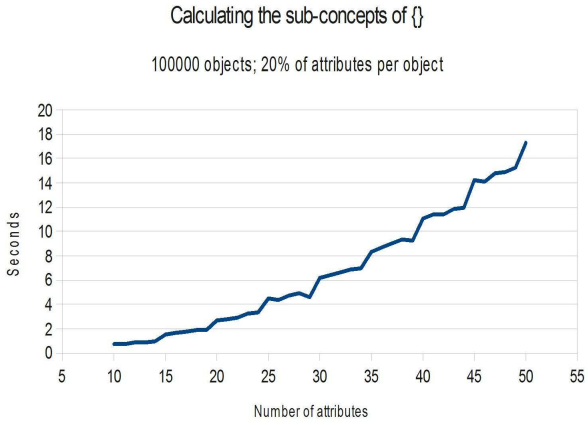
$$\frac{(|I| - |B|)(|I| - |B|)}{2} * O(c(B + X))$$

which leads to

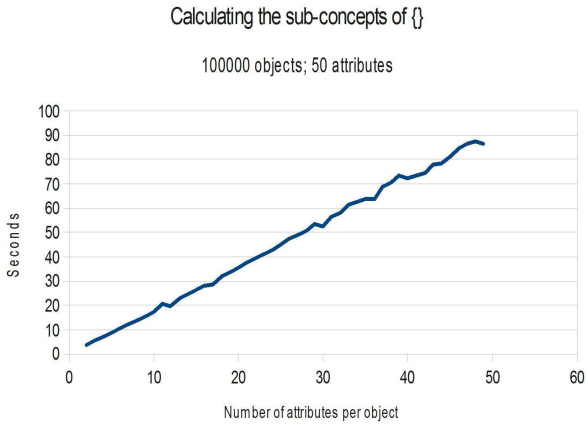
$$O((|I| - |B|)^2 * O(c(B + X)))$$

using the big O notation.

This has to be compared with $O(|I|^2 * |O|)$ of the Bordat's algorithm. In addition the cost $O(c_(B + x))$ of computing the cardinality of objects satisfying the required properties can be based on multiple keys and robust algorithms used in databases that do not need to load all data for computing a cardinality [5].



(a) 100.000 objects and attributes varying from 10 to 50, each object randomly described by 20 % of the attributes

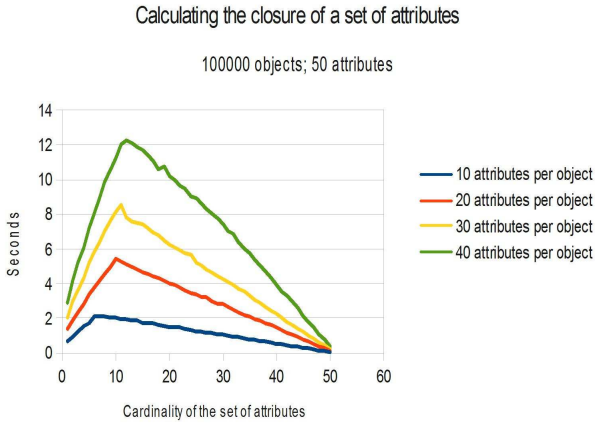


(b) 100.000 objects and 50 attributes, each object described by random attributes varying from 2 to 49

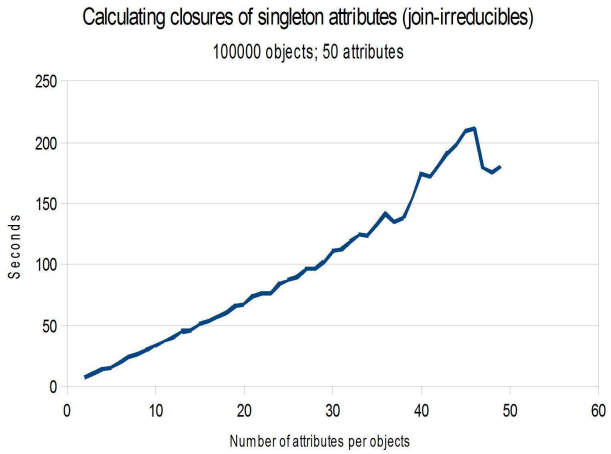
Fig. 1. Calculating the immediate successors of ∅

5 Experimentations

In the experiment, we use a dataset composed of 100.000 objects described by a random set of attributes varying from 10 to 50, each objects described by a random set of attributes. The dataset is stored in a database MySQL 5.5.17. We have implemented our algorithms using PhP 5.3.8 using a laptop with 8 processors clocked at 1.73GHz and 8Gb of memory. The counting of objects



(a) Average time of a closure generation for attributes varying from 1 to 50 with 100.000 objects and 50 attributes, each object described by 10, 20, 30 then 40 random attributes



(b) closure of attributes with 100 000 objects, 50 attributes, each object described by random attributes from 2 to 49

Fig. 2. Closures computation

satisfying a set of properties is realised by the SQL request comparing indexes with a constant:

```
select count (*) from att1=1 and att2=1
```

We compare the processing time of our algorithms in the following cases:

Immediate successors generation: we compute the immediate successors of the bottom concept in the two following cases:

1. 100.000 objects and attributes varying from 10 to 50, each object randomly described by 20% of the attributes (see Fig. 1(a))
2. 100.000 objects and 50 attributes, each object described by random attributes varying from 2 to 49 (see Fig. 1(b))

Closure generation: We compute closures in the following cases:

1. We compute the average time of a closure generation for attributes varying from 1 to 50 in the following case, with 100.000 objects and 50 attributes, each object described by 10, 20, 30 then 40 random attributes. (see Fig. 2(a))
2. We compute the closure of a singleton attribute with 100 000 objects, 50 attributes, each object described by random attributes from 2 to 49.

Trends of our results are consistent with theoretical complexities of the algorithms. Deporting a part of the calculation in the SQL engine, we believe that the index dedicated to counting object can improve performance. This results renforces those obtained in [5] on the efficience of the key-indexation techniques in SQL. In addition, new technologies related to cloud computing will also divide the workload of the SQL engine on demand.

6 Conclusion

In this paper, we described two basic algorithms for browsing in a concept lattice with limited objects access. By separating the counting from the rest of the algorithm, new systems for exploring concept lattices can now rely on optimization algorithms used in relational databases. If the tests we will realize on PostgreSQL and MySQL databases are successfull in terms of manipulating a huge amounts of data, we plan to propose a library for extending content management system such as Joomla!.

References

1. M. Barbut and B. Monjardet. *Ordres et classifications : Algèbre et combinatoire*. Hachette, Paris, 1970. 2 tomes.
2. G. Birkhoff. *Lattice theory*. American Mathematical Society, 3d edition, 1967.
3. N. Caspard and B. Monjardet. The lattice of closure systems, closure operators and implicational systems on a finite set: a survey. *Discrete Applied Mathematics*, 127(2):241–269, 2003.
4. B.A. Davey and H.A. Priestley. *Introduction to lattices and orders*. Cambridge University Press, 2nd edition, 1991.
5. C. Demko and K. Bertet. Generation algorithm of a concept lattice with limited object access. In *Proc. of Concept lattices and Applications (CLA'11)*, pages 113–116, Nancy, France, October 2011.
6. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with TITANIC. *Data and Knowledge Engineering*, 42(2):189–222, August 2002.

Relational Data Exploration by Relational Concept Analysis*

Xavier Dolques[1], Marianne Huchard[2], Florence Le Ber[3], and Clémentine Nebut[2]

[1] INRIA, Centre Inria Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes, France, xavier.dolques@inria.fr
[2] LIRMM, Université de Montpellier 2 et CNRS, Montpellier, France, first.last@lirmm.fr
[3] LHYGES, Université de Strasbourg/ENGEES, CNRS, Strasbourg, France florence.leber@engees.unistra.fr

Abstract. Relational Concept Analysis [4] is an extension to FCA considering several contexts with relations between them. Often used to extend the knowledge that can be learned with FCA, RCA also meets the issue of combinatorial explosion. The initial specification of RCA implies a monotonic growth of the number of concepts and an exhaustiveness of all the concepts that can be obtained when a fixed point is reached. In this position paper we propose a different specification of RCA that permits an interactive exploration of the data by letting the choice of the user for each step. This change will permit to handle richer relational data in a more flexible way by restraining the relations explored at each step hence reducing the number of created concepts.

1 Introduction

Relational Concept Analysis (RCA) [4] is based on iterative use of the classical Formal Concept Analysis algorithm to handle relational data: formal objects are described with formal attributes, and with their relationships with formal objects. Because RCA groups formal objects using relationships to formal objects at any distance, it often comes with a combinatorial explosion, and patterns of interest are difficult to extract from the huge set of built concepts. Various strategies can be used to cope with this complexity, including separating the initial formal object sets into smallest ones after a first analysis, or introducing queries [1]. Here we focus on the use of RCA to interactively explore data by letting the user choosing at each step of the iteration of FCA which contexts (formal and relational) he or she would like to use.

The context of this research is the FRESQUEAU project[4] which aims at developing new methods for studying, comparing and exploiting all the parameters available concerning streams and water areas. In this project, different

* This work was partly funded by french contract ANR11_MONU14.
[4] http://engees-fresqueau.unistra.fr/

approaches of knowledge discovery (including FCA) are tested and combined in order to better assess the ecological functioning of such hydrosystems.

In this paper we first outline the RCA process to highlight potential variation points that would promote exploration. Then we conclude with a short discussion.

2 The RCA algorithm

Algorithm 1 outlines the main steps followed by RCA to build groups of objects by considering attributes and object-object relations [4]. The input of RCA is a Relational Context Family $RCF = (K, R)$ composed of n object-attribute contexts $\mathcal{K}_i = (O_i, A_i, I_i)$, i in $1..n$, and m object-object contexts $\mathcal{R}_j$, j in $1..m$.

```
1: proc Multi-Fca( In: (K, R) a RCF,
2: Out: L array [1..n] of lattices)
3: p ← 0 ; halt ← false
4: for i from 1 to n do
5:   L^0[i] ← Build-Lattice(K_i^0)
6: while not halt do
7:   p++
8:   for i from 1 to n do
9:     K_i^p ← Extend-Rel(K_i^{p-1}, L^{p-1})
10:    L^p[i] ← Update-Lattice(K_i^p, L^{p-1}[i])
11:  halt ← ⋀_{i=1,n} Isomorphic(L^p[i], L^{p-1}[i])
```

Algorithm 1: The RCA process

For $\mathcal{R}_j \subseteq O_i \times O_j$, we call O_i the domain and O_j the range. The **initialization step** (Lines 4-5) consists in building, for all i in $1..n$, the lattice $\mathbf{L}^0[i]$ associated with the context $\mathcal{K}_i$.

At step p:

- **Extend-Rel** appends to $\mathcal{K}_i$ the relations obtained by scaling object-object relations for which $\mathcal{K}_i$ is the domain. The scaling consists in including the object-object relations as *relational* attributes. They are obtained using the concepts of the lattices of step $p-1$ and a scaling operator (*i.e.* $\exists$, $\forall$). For example, if the scaling operator $\exists$ is chosen for scaling a given relation $\mathcal{R}_j$, $\mathcal{R}_j$ columns are replaced by attributes of the form $\exists\mathcal{R}_j : C$, where C is a concept in the lattice built upon objects of the range of $\mathcal{R}_j$ at step $p-1$. An object o of the domain of $\mathcal{R}_j$ **owns** $\exists\mathcal{R}_j : C$ if $\mathcal{R}_j(o) \cap Extent(C) \neq \emptyset$.
- **Update–Lattice** updates the lattices of step $p-1$ in order to produce, for i in $1..n$, the lattice $\mathbf{L}^p[i]$, associated with $\mathcal{K}_i$ concatenated to all scaled object-object contexts with $\mathcal{K}_i$ domain.

The algorithm stops when a fix-point is obtained: a lattice family isomorphic to the lattice family obtained at the previous step is obtained and leaves unchanged concept extents.

The advantage of such a process is that the obtained concepts have in their intent relations to other concepts in addition to classical attributes. Those relations permit the extraction of patterns built from several interconnected concepts as shown in [2] and [3] that could not be easily obtained with the classical process of Formal Concept Analysis.

However, one problem of such a process is the potential difficulty to apprehend the result. In past work in the domain of Model Driven Engineering, data extracted from models of medium size have been easily handled by RCA. Nevertheless in a context of data mining the data are of a different scale. Especially when only small patterns are needed while many relations connect the objects and these relations form a cyclic entity-relationship diagram, the result will appear hard to understand by a human due to the number of concepts to consider simultaneously and the computation time will be considered as a handicap. In such cases, we think it will be more practical to have a kind-of exploratory approach.

Table 1 shows main possible variations on the algorithm to go towards an exploratory approach. We have enumerated the variation points of the algorithm that could affect the result by changing the contexts considered at each step. We have proposed for each variation point an alternative scenario from the process previously described that involves the user by asking him or her to perform selections. All those variations or only a subset of them can be applied depending on the granularity needed.

Table 1. Variations for the exploratory approach

initialization step, L4-5	Build lattices for selected object-attribute contexts concatenated to selected object-object contexts.
Extend-Rel, L9	Rather than using all relations and scaling all object-object relations, select a subset of the RCF and scaling operators for each selected object-object context. Note: lattices for ranges of the selected object-object relations should have been calculated in a previous step (not necessarily $p-1$). At this step, object-attribute contexts can also be selected and the corresponding lattice can be built.
Update–Lattice, L10	Only the lattices for the selected relations are updated.
halt, L11	The decision is left to the expert when to stop (or the fix-point is obtained)

3 Conclusion and discussion

In this position paper, we have outlined an exploratory approach for assisting the use of Relational Concept Analysis in a way that would better fit a data mining process. We have several motivations for disturbing the original RCA process: to go faster to a relevant result by calculating less lattices (preferably lattices of

interest), to cope with the inherent complexity of mining relational data, or to let the expert guiding the discovery process based on his/her intuition and the knowledge patterns that appear on-the-fly. In our current approach, the data are given by experts, so we don't use exploration in the sense of [5], unless the data exploration.

Many questions are raised by this way of extracting concepts from relational data. Initialization of the process has an impact for the later discovered structures. It can accelerate the process, if the selected object-object relations contain the main information for the expert, or reversely, it can discard the expert from the relevant information. Nevertheless, the most serious problem comes from the fact that going step-by-step leads to a non-monotonic concept construction and one could build several cases where the process diverges (iterates between recurrent configurations). In the original RCA process, when the fix-point is attained, lattices of the two last steps are isomorphic, thus when a concept references another through a relational attribute, the latter can be found in the same step appropriate lattice. But in the exploratory process we propose, when a concept references another through a relational attribute, the latter is in a lattice of a previous step and may itself reference a concept in a previous step, etc. We should find solutions for presenting the expert information easy to interpret these situations. Nevertheless, we think that such an exploratory approach should be more practical than the "brute force" that iterates until the fix-point and gives results that an expert will hardly understand.

References

1. Azmeh, Z., Huchard, M., Napoli, A., Hacene, M.R., Valtchev, P.: Querying relational concept lattices. In: Proc. of the 8th Intl. Conf. on Concept Lattices and their Applications (CLA'11). pp. 377–392 (2011)
2. Dolques, X., Huchard, M., Nebut, C.: From transformation traces to transformation rules: Assisting model driven engineering approach with formal concept analysis. In: Supplementary Proceedings of ICCS'09. pp. 15–29 (2009)
3. Dolques, X., Huchard, M., Nebut, C., Reitz, P.: Fixing generalization defects in UML use case diagrams. In: CLA'10: 7th International Conference on Concept Lattices and Their Applications. pp. 247–258 (2010)
4. Huchard, M., Hacène, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Ann. Math. Artif. Intell. 49(1-4), 39–76 (2007)
5. Rudolph, S.: Relational exploration: combining description logics and formal concept analysis for knowledge specification. Ph.D. thesis, Dresden University of Technology 2006 (2006)

Let the System Learn a Game: How Can FCA Optimize a Cognitive Memory Structure

William Dyce, Thibaut Marmin, Namrata Patel, Clement Sipieter, Guillaume Tisserant, and Violaine Prince

University Montpellier 2 and LIRMM-CNRS
Montpellier, France
{William.Dyce,Thibaut.Marmin,Namrata.Patel,
Clement.Sipieter}@etud.univ-montp2.com
{Tisserant,Prince}@lirmm.fr

Abstract. The goal of this article is to study the contribution of FCA (Formal Concepts Analysis) to (1) optimize (2) organize (3) discover new concepts or a better operation of the semantic memory of an Artificial Intelligence (AI) system based on a cognitive approach. The system has been applied to game modeling (here the Reversi board game), since games are a very good experimental field for performance evaluation. After describing the COGITO project, which tries to assess the pros and cons of cognitive modeling over pure operational but non explicative paradigms in games modeling, the paper stresses out the benefits of FCA in providing a better abstraction, and a more reliable way to handle conflictual knowledge.

Key words: Artificial Intelligence, Cognitive Modeling, Games, Semantic Memory, Formal Concepts Analysis

1 Introduction

One of the old dreams of Artificial Intelligence (AI) was to substitute humans with AI systems, in most of the chores involving problem solving. During the past fifty years, two methodological tracks have been extensively explored: Either imitating human behavior, a trend that naturally leads researchers to mimic the human cognitive structure, seen as the outcome of a natural selection [9]; Or definitely assessing that humans and computers are utterly different, and designing algorithms fitted to machines, thus bypassing human skills in problem solving [12]. If the first trend seems nowadays set aside because of its too many failures, this paper attempts to revive some of its claims, by constraining the project to a very simple task. This task, a REVERSI board game [7], has interesting basic properties:

– In a cognitive approach, games require different mechanisms: Capturing an input, trying to map it with the present memory state, learning it if new, and exploiting the integrated shape through reasoning when playing a new game. Thus, the behavior of a cognitive-based system could easily be tracked in its different steps.

– A totally different approach, the Minimax (also called the Von Neuman theorem, [11]), has given very good results. However, the Minimax is a way to win in a zero sum play, not a way to learn or to understand.
– This situation enables the evaluation of the pros and cons of a cognitive approach, versus a pragmatically performant but non explicative method, for a given task (even if more or less biased).

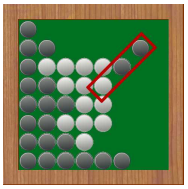


Fig. 1. A noteworthy pattern on a Reversi Board using the *aligned* predicate

This paper describes a part of a more extensive project named COGITO (both a research team, and an implemented software), restricted to the management and operation of the semantic memory , and the mechanisms that acquire (i.e. learn) or exploit (i.e. play) the knowledge required to learn to play a Reversi game.These aspects are implemented and functional. The goal of this article is, after describing the founding assumptions and the selected cognitive model, to study the contribution of FCA (Formal Concept Analysis) to (1) optimize (2) organize (3) discover new concepts or a better operation of the memory.

2 Designing a Reversi Board Game and Player

Figure 1 shows a Reversi board, with its black and white pieces. The aim of the game is to transform the adversary's pieces into one's own by placing the piece in such a way that it blocks the other's expansion. Thus, the play relies on noteworthy patterns that help the player develop winning strategies. There is a very scarce literature in AI applied to Reversi. In fact, a more modern and Japanese version, named Othello, has much more interested researchers. Rosenbloom [8] was the first to implement an Othello program (IAGO). Then, Lee and Mahajan have enhanced the program performances in their BILL program [4]. Another software, Logistello, has been developed by Buro, [1] who has further provided asurvey of Othello evolution [2]. All implementations were based on minimax evaluation functions. Later on, the game has been modeled with neural networks by [3], as a tentative approach to introduce cognitive-based models. Our attempt is the first to step from a performance-based software into a reasoning-based one.

2.1 The Game Requirements and their Modeling

In a computational framework, the play has been modeled with **predicates** that are the founding elements which compose these noteworthy patterns. The

retained basic predicates are the following:

1. $isMine(x)$: For the system, its own pieces
2. $isOpp(x)$: The adversary's pieces
3. $isEmpty(x)$: A position on the board which can possibly be occupied by a next move
4. $isEdge(x)$: A noteworthy position on the edge of the board. The piece that occupies it is harder to take.
5. $isCorner(x)$: Also a noteworthy position.
6. $near(x, y)$: Defines neighborhood. Might lead to a capture, if x are not of the same color as y.
7. $aligned(x, y, z)$: Three pieces on a same line, either vertical, horizontal, or even a diagonal. Allows to capture the two other pieces, if z is not of the same color as x and y.

To implement the game, one needs to:

- Acquire the board 'state', also called the **board configuration**. It requires the coordinates of all pieces, and which predicates each piece instantiates.
- Map the present board to a set of stored noteworthy patterns that express playing strategies.
- Choose the best and thus perform a **move**.
- Learn moves from the other player in order to enhance the system abilities.

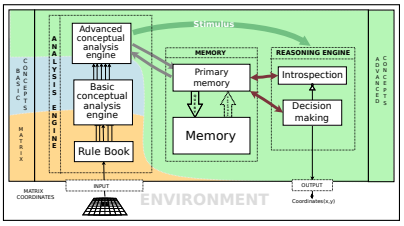


Fig. 2. The General Schema of the Implemented Cognitive Structure

2.2 The Implemented Cognitive Structure: Memory and Reasoning

The theoretical computational model underlying the design of such a requirement set is provided in figure 2. The board is described as a matrix, and is transmitted, from the *environment* to the *Rulebook* module, through an I/O module. The latter determines all the possible moves, and generates the set of all resulting matrices, to be transmitted to the *Basic Conceptual Analyzer*. This package transforms the matrices into a logical set of first order formulas, using the basic predicates defined above. The outcome is a set of facts in a logical format, transmitted to the *Advanced Conceptual Analyzer*. The latter maps the possible patterns of the board with the already stored noteworthy patterns. Then, it

launches the *reasoning module* which chooses between the possible moves, according to the set of board configurations and recognized noteworthy patterns. This choice is based on an evaluation associated with the pattern, representing the number of times it has figured in a winning game (in the form of a 'probability of winning' with the appropriate formula). This cognitive structure schema has been largely inspired from the 'artificial consciousness model' in [10]. The latter involves a much larger set of elements and relationships. The COGITO project work has mostly focused on the memory and reasoning parts of the model. As seen here, the primary memory is a temporary buffer that stores the results of perceived inputs (short term memory). The memory module contains two other parts: An *episodic memory*, storing games and moves as they have been played during different sessions, and the *semantic memory*, keystone of this contribution. Both have exactly the same structure, and the episodic memory content is 'appended' to the semantic memory.

3 The Semantic Memory Structure

3.1 Memory as a Graph Structure

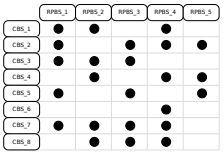


Fig. 3. The Semantic Memory Matrix

The semantic memory stores:
(1) *Objects*, that represent board configurations met during different games. These objects are implemented as classes named **Complete Board States** or **CBS**.
(2) *Attributes*, for those noteworthy patterns added, all along, by the reasoning module introspective part, and named **Relevant Partial Board States** or **RPBS**.
(3) Relationships between boards and patterns, i.e. between CBS and RPBS.
Figure 3 is a representation of the semantic memory content. As such, this has naturally led us to consider two possible approaches for shaping and formalizing the semantic memory:
(1)A bi-part graph, where objects and attributes are nodes, and their edges standing for their mutual relationships, such as in figure 4. The prefix 'master' seen in this graph allows typing any graph (from the episodic memory), appended to the stored parts of the long term semantic memory (Figure 5 shows how the semantic memory is upgraded with parts coming out of the episodic memory). The root node suggests here an artificial referential element, neutral to the relationship 'edge'.
(2) A conceptual model obtained after applying FCA.
The graph representation has been implemented here with a graph oriented

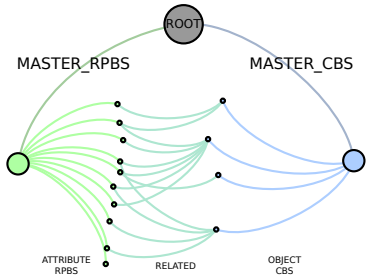


Fig. 4. A Graph Representation of the Semantic Memory

Fig. 5. Appended Graphs of the Episodic Memory, Completing the Semantic Memory Graph

DBMS (an open source system called Neo4j), and exploited. It works quite well in most of the cases.

However, observation has shown the following liabilities, that were transformed into requirements for an FCA modeling:

- The abstraction level is quite low, and still too close to the operational requirements of the game.
- When reasoning on patterns as pure attributes, any composition of patterns inherits the valuation of its members. For instance, if two 'winning' RPBS, when associated, could generate a conflict, the present approach would not detect it.
- It is possible that the information appended from the episodic memory and some already existing parts of the semantic memory, turn out to be redundant. The present model does not prevent such a situation, neither does it cure it.

3.2 The Contribution of FCA to the Semantic Memory Organization

FCA (Formal Concepts Analysis) [6] helps organizing and structuring information presented as a collection of objects and their properties. Figure 3 shows that the semantic memory content is a very good candidate for such a design. Thus, it has been performed on the CBS/RPBS matrices presented in Figure 3, using Concept Explorer (Conexp, [13]) an open source concept lattice builder. A recent work on a neighboring application, related to semantic neural decoding [5] has encouraged such an attempt. Human cognitive structures are neural, and an imitative model, such as our Cognitive Semantic Memory, would probably benefit from the same achievements. The discussed improvements are those described in the following subsections.

Reducing Redundancy, Optimizing Decision Making, Evaluating Patterns Quality and Discovering New Patterns In Figure 6, three concepts introduce more than five noteworthy patterns (RPBS). This helps to reduce redundancy, as mentioned previously, by merging these patterns into one, without

any loss of information on the whole set of acquired data. Let us note that FCA concepts are sets of CBS sharing the same properties. This means that games could be put into categories, and a classification of winning games and their hierarchy, can be extracted from this work (see next subsection).

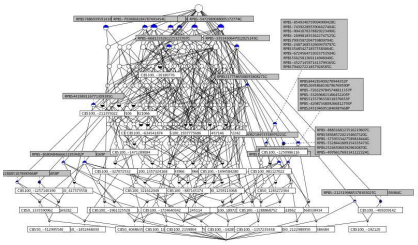


Fig. 6. An overview of the Concepts Lattice: Boards are Objects, and Patterns, Attributes

Also, a hierarchy among the RPBS seems to appear. This might lead to a more intelligent search on the matching RPBS (in the reasoning module of Figure 2) by reducing the number of homomorphisms applied to each board. Thus, the decision making might take less time, without reducing the number of RPBS in the search space.

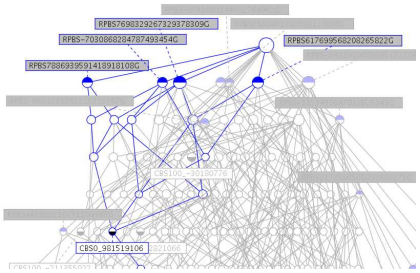


Fig. 7. noteworthy Patterns emerging as Common Parts of Concepts, i.e. Game Boards

The RPBS introduced in the parent concepts of a common concept (introducing at least one CBS) have potentially a common subpart, that could be extracted as a new (and efficient) RPBS (see Figure 7).
Also, RPBS have been weighted, in the COGITO system, according to their presence frequency in winning, respectively losing games. We have tried to build lattices based on this feature. Figure 8 shows the organization of the winning games. Such a piece of information is crucial since it might help modifying the RPBS weights.

Discovering New Rules and Strategies Moreover, and unexpectedly, Conexp has helped us find rules featured as follows, that might be caused by inclu-

sions of RPBS:

$$r < n > RPBS_{id}(w) =>< m > RPBS_{id1}, RPBS_{id2}, ...RPBS_{idk} \quad (1)$$

where:

1. r is the rank of the rule. The better the rank, the more reliable the rule.
2. $< n >$ represents the number of times the RPBS identifier ($RPBS_{id}$) in the hypothesis appears in a CBS.
3. m is the number of times the conclusion is found.
4. w represents the confidence associated to the rule. if w is equal to 100, it means that each time the $RPBS_{id}$ is found, there is a 100 percent chance that it is followed by the $RPBS_{id1}, RPBS_{id2}, ...RPBS_{idk}$ of the conclusion. w is the representation of n/m. The following extract shows a few among those that have been found by the system.

This means that beyond the relationships between concepts (CBS) and attributes (RPBS), FCA helps discovering possible dependancies amongst attributes themselves, leading to a re-design of the noteworthy pattern notion.
Samples of Derived Rules

```
1 < 7 >   RPBS2451946951846987668P [100 \%]
   ==> < 7 > RPBS-429873480926681279 3P   RPBS5155796358318376653P
    RPBS-3326968131664522 05P  RPBS-72632978457488113 57P
      RPBS2695868336796769590P  RPBS844283409270944352P;
 [..]
58 < 20 > RPBS63684013935981136 86G  [95 \%]=>
 < 19 > RPBS6176995682082658 22G;
[..]
```

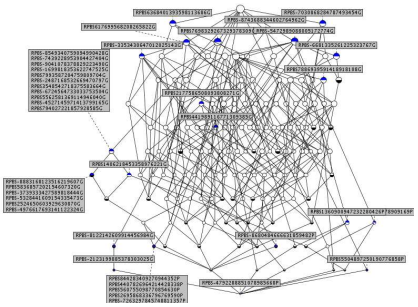


Fig. 8. A Lattice of the Winning Games and the Patterns they used

4 Conclusion

The experiment has shown that FCA can provide very interesting modifications to the initial memory structure. For the moment this step has not been automated, because we wanted to evaluate FCA added value: indeed, it has improved

the quality and reliability of the acquired knowledge. However, FCA must not be run on a learning system, since it would bias the learning step (when the system acquires new RPBS and CBS from games played against a human player). The general idea is to re-design the memory with FCA, and this time, automatically, but only after a reasonable number of games where the memory has more or less acquired elements, and to rerun FCA only until another quite large number of games have been played. During the game step, it would be valuable to store the ongoing CBS in a concept. Concepts can be 'weighted' with values expressing their reliability. Thus the ongoing CBS can inherit its parent concept present weight and, benefiting from the lattice structure, drastically reduce the number of possible RPBS (in the reasoning module). Also, it would be interesting to constantly check stabilization in learning, in order to build a sort of a final lattice, which will represent a stable and 'mature' state of the semantic memory. We also anticipate that the final number of concepts will also stabilize. A future experiment will be performed to determine the final lattice size.

References

1. Buro, Michael. "The Othello Match of the Year: Takeshi Murakami vs. Logistello", ICCA Journal, vol 20, 3, P.189-193 (1997).
2. Buro, Michael. The Evolution of Strong Othello Programs", in: Entertainment Computing - Technology and Applications, R. Nakatsu and J. Hoshino (ed.), Kluwer, p. 81-88 (2003).
3. Chong, S.Y., Tan M.K, White, J.D. "Observing the evolution of neural networks learning to play the game of Othello", IEEE Transactions on Evolutionary Computation, Vol 9, 3, p.240-251(2005)
4. Lee,K; Mahajan, S. "The development of a world-class Othello program". Artificial Intelligence, vol. 43, p. 21-36.(1990).
5. Endres, Dominik M.; Foldiak, Peter; Priss, Uta. "An Application of Formal Concept Analysis to Semantic Neural Decoding", Annals of Mathematics and Artificial Intelligence, Vol, 57, 3, Springer-Verlag, p. 233-248 (2010).
6. Ganter, Bernhard; Stumme, Gerd; Wille, Rudolf, eds, "Formal Concept Analysis: Foundations and Applications", Lecture Notes in Artificial Intelligence, no. 3626, Springer-Verlag. (2005)
7. A description of the Reversi Game. http: //en.wikipedia.org/wiki/Reversi.
8. Rosenbloom, P. "A world-championship level Othello program". Artificial Intelligence, vol. 19, p.279-320. (1982).
9. Sweller, John. "Instructional Design Consequences of an Analogy between Evolution by Natural Selection and Human Cognitive Architecture". Instructional Science, 32, 1, 9-31.Springer Netherlands (2004)
10. Tisserant, Guillaume; Maurin, Guillaume; Ndongo, Wand ; Villemot, Anthony. "Rapport sur une conscience artificielle". LIRMM-CNRS research Report.(2010)
11. von Neumann, John. Zur Theorie der Gesellschaftspiele , Mathematische Annalen, vol. 100, p. 295-320(1928).
12. Warwick, Kevin. "March of the machines: the breakthrough in artificial intelligence" Univ. of Illinois. (2004)
13. Yevtushenko Serhiy A. "System of data analysis "Concept Explorer"." (In Russian),Proceedings of the 7th national conference on Artificial Intelligence KII-2000, p. 127-134, Russia, (2000).

An approach to Semantic Content Based Image Retrieval using Logical Concept Analysis. Application to comicbooks.

Clément Guérin, Karell Bertet and Arnaud Revel

L3I, University of La Rochelle, France
{cguerin,kbertet,arevel}@univ-lr.fr

Abstract. In this paper, we present an ongoing work aiming to improve content based image retrieval performance with the help of logical concept analysis. Domain semantic is formalized and used instead of classical CBIR visual features. This is being applied to comicbooks using Sewelis.

Keywords: Comic Books, Description Logics, Semantic, Logical Concept Analysis, Content-Based Image Retrieval.

1 Introduction

Web search engines usually give poor results when searching in multimedia databases since they use the contextual web page, or the meta information attached to the multimedia objects. The semantic meaning that the user usually attaches to the content of the document is often very different from the text used for indexing the image (semantic gap). Content Based Image Retrieval (CBIR) has been proposed to search into huge unstructured image databases by giving an example of what the user is looking for instead of describing the concept it represents. Classically, visual features are extracted from the images and then compiled into a signature [1]. To perform the retrieval, a similarity function is computed to compare the index of the query to those of the collection. A ranking of the results is produced according to the similarity and shown to the users. To improve the quality of the retrieval, an interaction with the user, called relevance feedback [2], can be added. These techniques work pretty well in the context of searching visually similar images in unstructured image databases.

In this article, we are interested in CBIR in the context of comicbook databases. In this case, databases cannot be considered as unstructured anymore since images can be grouped in terms of panels, pages and volumes which are themselves associated with metadata concerning the author or the series they belong to. We would like to benefit both from the search facilities given by CBIR techniques with feedback and semantic information embedded in the structure of the comicbooks documents. To do such a thing, Logical Concept Analysis (LCA), an extension of Formal Concept Analysis (FCA) [3], is used through the Sewelis implementation [4]. We will first go through the presentation of our comicbook model and its transcription into LCA. Then we will explain how we can mix classical CBIR and LCA techniques together to enhance retrieval relevance.

2 Semantic Content Based Image Retrieval

2.1 Model description

Comicbooks have a natural hierarchical structure that can be formalized. They are made of pages which contain panels. These panels can eventually be gathered in strips[1] and contain different kind of objects, such as speech balloons, characters, free text, etc. Balloons can be of many kinds (dialogue, thoughts etc.).This knowledge can be used to deduce more information such as pieces of the storyline. Fig. 1 illustrates the model we propose to formalize the comicbooks domain. It has been described with more details in [5].

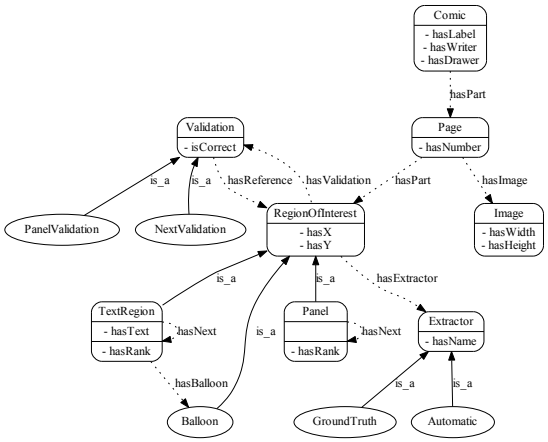


Fig. 1. Part of our model concepts hierarchy and their properties.

2.2 Heterogeneous and complex data integration

Some works [6–8] already enhanced the classical CBIR techniques with an ontology approach. The modelling was mainly focused on the description of segmented areas though. We would like to go further and use the full power of description provided by description logics.Indeed, the model presented previously is expressive enough to allow the retrieval of similar panels considering different characteristics like low-level image features, spatial relations or semantic information.

An input picture, picked from the database, being given, the system will not only be able to retrieve similar pictures based on the classical image characteristics (colors, shapes, textures...), but also based on the associated semantic and the knowledge that could have been learnt previously. Considering that the query is a Panel instance, the search can focus on:

(1) The panel's characteristics (i.e. data properties of a Panel object). This could be its rank, its shape, its size, its position, its shot type, its view angle,

[1] A strip is defined as an horizontal sequence of panels. Traditionally, a strip is made of 1 to 6 panels and a page can contain up to 4 stacked strips.

etc. Images of a *very close shot of a character's face* or a *landscape picture of a valley being at the top of a page* can be examples of queries.

(2) The panel's relations (i.e. object properties). Properties of objects related to the query panel can be used as well as its own characteristics. Therefore, there are two directions to look at from a panel point of view.

- The search can focus on what is *inside* the panel, like similar amount of objects in a scene (a dialogue between two characters for instance) or related text content. The retrieval process can also rely on objects contained in the panel, whether they are identified or not. Assume that the query picture contains an identified character A whose visual signature is defined by the set of features X. The system will not only look for panels containing an instance of A, but also for those showing a spatial region matching X.
- Outside: the search can focus on panels sharing page's or comic's characteristics (such as author, style, etc.)

These kinds of retrieval angles are not mutually exclusive and it is very possible to combine two or more of them in order to narrow the result set. The search possibilities are only limited by the completeness of the description.

2.3 Sewelis integration

In databases, information retrieval is classically performed by request queries expressed in a specific request language, as SQL for example. However, the more refined is the search, the more sophisticated is the request. Some information retrieval systems offer a simpler search refinement by navigation in a predefined static data structure, where each navigation step proposes to the user a more refined query answer. For example, file systems can be considered as an information retrieval system where data is organized in a static tree structure. A new approach of information retrieval, both by request and by navigation in a Galois lattice structure [9], has been proposed in [10, 11].

The concept lattice is a rich and flexible navigation structure automatically derived from data, and can therefore be considered as a dynamic and complete space search enabling data description while preserving its diversity. Querying and navigation can be freely combined: to each user request corresponds a concept of the lattice as answer ; the user can then improve its search either by amending its request, or by on-line browsing around the concept in the lattice structure. Such an approach was already proposed, for example in [10] with the logical information systems (LIS) and has been implemented in Sewelis [4].

Sewelis is used to load the comics' ontology and to create a bound between the model and a concept lattice. The objects of the lattice match the classes of the model, the attributes are their properties and each concept stands for a set of classes' instances sharing the same properties. It is then possible to navigate all the way to any concept, using the flexible navigation structure provided by the concept lattice.

2.4 Application

Let us illustrate this with a simple example. Let say we have a query panel and we want to retrieve the strip it is coming from. While it only takes a quick look

to a human being to find the answer, it is not something obvious for a machine, the *strip* concept not even being part of the model. Classical CBIR methods, based on a similar visual features criterion, are helpless in that case. However, if the knowledge related to the panels and their inside/outside relatives is used, it becomes possible to return results that can be justified by the system and iteratively refined with the relevance feedback brought by the user. Concerning this request, the page number of the panel will first be considered (outside panel's relation) in order to focus on panels coming from the same page. Then, the y-axis value of its centroid will be selected and only panel's whose centroid corresponds to the same y-value, within a predefined delta, will be kept. Finally, the *hasNext* [5] relation can be used to sort output panels in order to rebuild the strip.

3 Conclusion and perspectives

This paper has presented an ongoing work about a Semantic Content Based Image Retrieval system applied to comic books. The final aim would be to provide a complete system that would be able to (1) retrieve resources similar to a query, based on the amount of mutual properties they share and the significance of these properties guided by the user relevance feedback, and (2) explain to the user why a returned resource is considered to be relevant to the query.

References

1. R.C. Veltkamp. Content-Based Image Retrieval System: A Survey. In *Technical report, University of Utrecht*, 2002.
2. M.E.J. Wood, N.W. Campbell, and B.T. Thomas. Iterative refinement by relevance feedback in content-based digital image retrieval. In *ACM Multimedia 98*, pages 13–20, September 1998.
3. B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations*. Springer Verlag, Berlin, 284 pages, 1999.
4. S. Ferré and A. Hermann. Semantic search: reconciling expressive querying and exploratory search. In *Proceedings of the ISWC'11*, 2011.
5. N. Tsopze, C. Guérin, K. Bertet, and A. Revel. Ontologies et relations spatiales dans la lecture d'une bande dessinée. In *IC*, pages 175–182, Paris, 2012.
6. P. Stanchev, D. Green Jr, and B. Dimitrov. High level color similarity retrieval. 2003.
7. Y. Liu, D. Zhang, G. Lu, and W.Y. Ma. Region-based image retrieval with perceptual colors. *Advances in Multimedia Information Processing-PCM 2004*, pages 931–938, 2005.
8. V. Mezaris, I. Kompatsiaris, and M.G. Strintzis. An ontology approach to object-based image retrieval. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–511. IEEE, 2003.
9. G. Birkhoff. *Lattice theory*, volume 25. American Mathematical Society, 418 pages, third edition, 1967.
10. S. Ferré and O. Ridoux. An introduction to logical information systems. *Information Processing & Management*, 40(3):383–419, 2004.
11. C. Carpineto and G. Romano. *Concept data analysis*. Wiley Online Library, 2004.

CLASSIFICATION REASONING AS A MODEL OF HUMAN COMMONSENSE REASONING

Xenia A. Naidenova

Military Medical Academy, Saint-Petersburg, Russian Federation

ksennaidd@gmail.com

Abstract. In this article, it is proposed to consider classification reasoning based on inducing and using implicative dependencies as a model of commonsense reasoning. The main concept of this reasoning is a good classification test considered as a formal concept of the FCA. The Galois lattice is used for constructing good classification tests. Special rules are determined for constructing Galois lattices over a given context. All the operations of lattice construction take their interpretation in human mental acts.

Keywords: Commonsense reasoning, Classification test, Machine Learning, Inductive-deductive reasoning, Formal Concept Analysis.

1 Introduction

The symbolic methods of machine learning work on objects with symbolic, Boolean, integer, and categorical attributes. From this point of view, these methods can be considered as the methods of mining conceptual knowledge or the methods of conceptual learning. Currently the theory of symbolic machine learning is not recognized as a model of classification reasoning, although precisely this reasoning constitutes an integral part of any mode of reasoning (1, 2). The sole exception to this is the DSM method of hypothesis generation developed by V.K. Finn (3) and based on simulating inductive reasoning rules revealed in human thinking by D. S. Mill (1). There is also a tradition to consider induction separately from deduction. Classification task of mining hypotheses distinguishing and describing classes of a given object classification is conventionally solved separately from hypothesis's application except the deductive-inductive integrated model developed by Zakrevskij (4) and based on representation of data and knowledge in Boolean space of attributes.

However the role of classification in human reasoning is enormous. Classification, as a process of thinking, performs the following global operations: 1) forming knowledge and data contexts adequate to a current situation of reasoning; 2) reducing the domain of the search for a solution of some problem; 3) generalizing or specifying object descriptions; 4) interpreting logical expressions on a set of all thinkable objects; 5) revealing essential elements of reasoning (objects, attributes, values of attrib-

utes etc); 6) revealing the links of object sets and their descriptions with external contexts interrelated with them. This list can be continued.

Reasoning requires a lot of techniques related to increasing its efficiency such as valuation, anticipation, making hypotheses, generalization and specification. One of the important techniques is decomposition of the main problem into sub-problems. It implies using the following operations: choosing sub-problems, ordering sub-problems (ordering arguments, attributes, objects, variables, etc.), optimizing sub-problem selection, and some others. The most familiar examples of sub-problem ordering are so called tree-like scanning and level-wise scanning methods. Some interesting variations of selecting sub-problems are the choice of a more flexible sub-problem, for example, one with minimal difference from a previous sub-problem and a sub-problem with minimal possible number of new solutions. Intermediate results of reasoning are used for decreasing or locally bounding the number of sub-problems.

We limit our consideration of classification reasoning to a special class of logical reasoning based on mining and using conceptual knowledge the elements of which are objects, attributes (values of attributes), classifications (partitions of objects into disjoint blocks), and links between them. If we take into account that implications express relations between concepts (the object $\leftrightarrow$ the class, the object $\leftrightarrow$ the property, the property $\leftrightarrow$ the class), we can assume that schemes of mining and applying implications form the core of classification processes, which, in turn, form the basis of human commonsense reasoning.

Our approach is based on the concept of a good diagnostic test (GDT) for a given classification of objects (5). A good classification test has a dual nature: on the one hand, it is a logical expression in the form of implication or functional dependency, on the other hand, it generates the partition of a training set of objects equivalent to the given classification of this set or the partition that is nearest to the given classification with respect to the inclusion relation between partitions. Inferring good test allows in principle mining from data not only structures of formal concepts but also structures of classification ordered by the inclusion relation.

Mathematical structure for GDTs' construction is Galois's lattice. The formal model of classification as an algebraic lattice has been obtained in two independent ways. One way goes back to the work of great psychologist J. Piaget who introduced the concept of grouping (2) to explain methods of object classification developed by 7-11 years children. In this book, a conception of classification is given based on mutually coordinated operations on objects, classes of objects, and properties of objects.

The coordinated classification operations generate logical implicative assertions. The classification operations are connected with understanding the operations of quantification: "not all c are a", "all b are c", "no b are c", "some c are b", "some b are not a" and so on. The violation of the coordinated classification operations implies the violation of reasoning. Piaget J. shows that a key problem of personal understanding operational classification is the problem of understanding the inclusion relation. He adds that the lattice structure is the source of classification operations (2, pp. 195, 387-389).

The idea that classification is a lattice arose also from practical tasks of pattern recognition. In 1974, J. Shreider has described the classification algebra (6) as idempotent semigroup with the unit element. In 1974, N. Boldyrev advanced (7) the formalization of pattern recognition system as algebra with two binary operations of refinement and generalization defined by an axiom system including lattice axioms.

The paper is organized as follows: basic definitions are given in Section 2, Section 3 describes briefly a model of lattice construction as inductive-deductive commonsense or classification reasoning; some words of conclusion terminate this article.

2 Basic Definitions

IMPLICATIVE ASSERTIONS (logical rules of the first kind) describe regular relationships connecting together objects, properties and classes of objects. We consider the following forms of assertions: implication (a, b, $c \to$ d), forbidden rule (a, b, $c \to$ false (never), diagnostic rule (x, $d \to a$; x, $b \to$ not a; d, $b \to$ false), rule of alternatives (a or $b \to$ true (always); a, $b \to$ false), compatibility (a, b, $c \to VA$, where VA is the occurrence's frequency of the rule).

In our consideration, COMMONSENSE REASONING RULES (CRRs) are rules with the help of which implicative assertions are used, updated and inferred from instances. The deductive CRRs infer consequences from observed facts with the use of implicative assertions. An analysis of human commonsense reasoning shows that these rules are the following ones: modus ponens: "if A, then B"; A; hence B; modus ponendo tollens: "either A or B" (A, B – alternatives); A; hence not B; modus tollendo ponens: "either A or B" (A, B – alternatives); not A; hence B; modus tollens: "if A, then B"; not B; hence not A; generating hypothesis: "if A, then B"; B; A is possible.

The inductive CRRs are the canons formulated by John Stuart Mill (1): Method of Agreement, Method of Difference, Joint Method of Agreement and Difference, Method of Concomitant Changes, and Method of Residuum. These methods are not rules but they are the processes in which implicative assertions are generated and used immediately. Therefore inductive inferences are not separated from deductive ones.

Let $G = \{1, 2,\ldots, N\}$ be the set of objects' indices (objects, for short) and $M = \{m_1, m_2, \ldots, m_j, \ldots m_m\}$ be the set of attributes' values (values, for short). Each object is described by a set of values from M. The object descriptions are represented by rows of a table the columns of which are associated with the attributes taking their values in M (see, please, Table 1).

The definition of good tests is based on correspondences of Galois on $I = G \times M$ (8) and two relations $G \to M$, $M \to G$. Let $A \subseteq G$, $B \subseteq M$. Denote by B_i, $B_i \subseteq M$, $i = 1,\ldots, N$ the description of object with index i. We define the relations $G \to M$, $M \to G$ as follows: $G \to M$: $A' = \text{val}(A) = \{\text{intersection of all } B_i: B_i \subseteq M, i \in A\}$ and $M \to G$: $B' = \text{obj}(B) = \{i: i \in G, B \subseteq B_i\}$. Of course, we have $\text{obj}(B) = \{\text{intersection of all obj}(m): \text{obj}(m) \subseteq G, m \in B\}$.

Operations val(A), obj(B) are reasoning operations (derivation operators) related to discovering general features of objects and all objects possessing a given set of features.

We introduce two generalization operations: generalization_of(B) = B'' = val(obj(B)); generalization_of(A) =A'' = obj(val(A)). These operations are actually closure operators (8). A set A is closed if A = obj(val(A)). A set B is closed if B = val(obj(B)). For $g \in G$ and $m \in M$, $\{g\}'$ is denoted by g' and called **object intent**, and $\{m\}'$ is denoted by m' and called **value extent**.

The generalization (specification) operations are usual mental acts. Suppose that somebody has seen two films with the participation of Gerard Depardieu. After that he tries to know all the films with his participation. Suppose that one can know that Gerard Depardieu acts with Pierre Richard in several films. After that he can discover that these films are the films of the same producer Francis Veber.

For representing a classification, we use factually the way proposed by S.O. Kuznetsov in (9) for the case when the set M is the set of attribute's values. Let a context $K = (G, M, I)$ be given. In addition to values of M, a target value $\omega \notin M$ of an attribute is considered. The set G of all objects is partitioned into two subsets: the set G_+ of objects having property ω (positive objects), the set G_- of objects not having property ω (negative objects). We have $K = K_+ \cup K_-$, where $K_+ = (G_+, M, I_+)$, $K_- = (G_-, M, I_-)$, $G = G_+ \cup G_-$ ($G_- = G \setminus G_+$). Diagnostic test is defined as follows.

Definition 1. **A diagnostic test** for G_+ is a pair (A, B) such that $B \subseteq M$ (A = obj(B) $\neq \varnothing$), $A \subseteq G_+$ and $B \not\subset$ val(g) & $B \neq$ val(g), $\forall g$, $g \in G_-$. Equivalently, obj(B) $\cap$ G_- = $\varnothing$.

In general case, a set B is not closed for diagnostic test (A, B), i. e., a diagnostic test is not obligatory a concept of FCA. This condition is true only for the special class of tests called 'maximally redundant ones'.

Definition 2. A diagnostic test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for G_+ is **maximally redundant** (GMRT) if obj($B \cup m$) $\subset$ A, for all $m \notin B$ and $m \in M$.

Definition 3. A diagnostic test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for G_+ is **irredundant** if any narrowing $B^* = B \setminus m$, $m \in B$ implies that (obj(B^*), B^*)) is **not a test** for G_+.

Definition 4. A diagnostic test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for G_+ is **good** if and only if any extension $A^* = A \cup i$, $i \notin A$, $i \in G_+$ implies that (A^*, val(A^*)) is **not a test** for G_+.

If a good test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for G_+ is irredundant, then any narrowing $B^* = B \setminus m$, $m \in B$ implies that (obj(B^*), B^*)) is **not a test** for G_+. If a good test (A, B), $B \subseteq M$ (A = obj(B) $\neq \varnothing$) for G_+ is maximally redundant, then any extension $B^* = B \cup m$, $m \notin B$, $m \in M$ implies that (obj($B^* \cup m$), B^*) is **not a good test** for G_+.

Definition 5. Let t be a set of values such that (obj(t), t) is a test for a given set of objects. We say that the value $m \in M$, $m \in t$ is essential in t if (obj($t \setminus m$), ($t \setminus m$)) is not a test for a given set of object.

Definition 6. Let s be a subset of objects belonging to a given positive class of objects; assume also that (s, val(s)) is not a test. The object t_j, $j \in s$ is said to be an essential in s if ($s \setminus j$, val($s \setminus j$)) proves to be a test for a given set of positive objects.

To illustrate using essential values and generalization operations in the process of good tests' generation, we consider a partition of objects in Table 1 into positive and negative ones. Let $G(+)$ be equal to {4,5,6,7,8} and $splus(m)$ = obj(m) $\cap$ $G(+)$, $m \in T$. The value *'Red'* corresponds to a test for positive objects because obj(*Red*) =

splus(*Red*) $\subseteq$ *G*(+). Delete '*Red*' from consideration. The value *'Tall'* is essential one in object 7 and does not correspond to a test: obj(*Tall*) = {3,4,5,7,8} $\neq$ *splus*(*Tall*). The projection of the value '*Tall*' on the set of positive objects is in Table 2. Here *splus*(*Bleu*) = {5,7,8}, val(*splus*(*Bleu*)) = '*Tall Bleu*', obj(*Tall Bleu*) = *splus*(*Tall Bleu*), hence '*Tall Bleu*' corresponds to a test for Class 2. We have also that '*Tall Brown*' corresponds to a test but not a good one. We delete '*Bleu*' and *'Brown*' from the projection as shown in Table 3.

Table 1. Example of a data classification

Index ofexample	Height	Color of hair	Color of eyes	Class
1	Low	Blond	Bleu	1
2	Low	Brown	Bleu	1
3	Tall	Brown	Hazel	1
4	Tall	Blond	Hazel	2
5	Tall	Brown	Bleu	2
6	Low	Blond	Hazel	2
7	Tall	Red	Bleu	2
8	Tall	Blond	Bleu	2

Table 2. The projection of the value 'Tall' on objects of *G*(+)

Index of example	Height	Color of hair	Color of eyes	Class
4	Tall	Blond	Hazel	2
5	Tall	Brown	Bleu	2
7	Tall		Bleu	2
8	Tall	Blond	Bleu	2

Table 3. The reduction of the projection of the value 'Tall' on objects of *G*(+)

Index of example	Height	Color of hair	Color of eyes	Class
4	Tall	Blond	Hazel	2
5	Tall			2
7	Tall			2
8	Tall	Blond		2

Now rows 5 and 7 do not correspond to tests for Class 2 and they can be deleted. The intersection of the remaining rows of the projection is '*Tall Blond*'. We have that obj(*Tall Blond*) = {4,8} $\subseteq$ *G*(+) and (obj(*Tall Blond*), *Tall Blond*) is a test for Class 2. As we have found all the tests for Class 2 containing '*Tall*' we delete '*Tall*' from the objects of this class. Return to Table 1. We can delete rows 5, 7, and 8 because they do not correspond to tests for Class 2: value *Tall* is essential one in these rows. The intersection of the remaining objects of Class 2 gives a test (obj(*Blond Hazel*), *Blond Hazel*) because obj(*Blond Hazel*) = *splus*(*Blond Hazel*) = {4,6} $\subseteq$ *S*(+).

3 Inferring good classification tests as commonsense reasoning

We shall consider two interconnected lattices OBJ = (2^G, $\cup$, $\cap$) = (2^G, $\subseteq$) and VAL = (2^M, $\cup$, $\cap$) = (2^M, $\subseteq$), where 2^G, 2^M designate the set of all subsets of objects and the set of all subsets of values, respectively; $s \in 2^G$, $t \in 2^M$. Inferring the chains of lattice

elements ordered by the inclusion relation lies in the foundation of generating all diagnostic tests: (1) $s_0 \subseteq \ldots \subseteq s_i \subseteq s_{i+1} \subseteq \ldots \subseteq s_m$ ($val(s_0) \supseteq val(s_1) \supseteq \ldots \supseteq val(s_i) \supseteq val(s_{i+1}) \supseteq \ldots \supseteq val(s_m)$) ; (2) $t_0 \subseteq \ldots \subseteq t_i \subseteq t_{i+1} \subseteq \ldots \subseteq t_m$ ($obj(t_0) \supseteq obj(t_1) \supseteq \ldots \supseteq obj(t_i) \supseteq obj(t_{i+1}) \supseteq \ldots \supseteq obj(t_m)$). The dual ascending and descending processes of lattice generation are determined as follows: (3) $t_0 \supseteq t_1 \supseteq \ldots \supseteq t_i \supseteq ti+1 \supseteq \ldots \supseteq t_m$ ($obj(t_0) \subseteq obj(t_1) \subseteq \ldots \subseteq obj(t_i) \subseteq obj(ti+1) \subseteq \ldots \subseteq obj(t_m)$) ; (4) $s_0 \supseteq s_1 \supseteq \ldots \supseteq s_i \supseteq s_{i+1} \supseteq \ldots \supseteq s_m$ ($val(s_0) \subseteq val(s_1) \subseteq \ldots \subseteq val(s_i) \subseteq val(s_{i+1}) \subseteq \ldots \subseteq val(s_m)$).

The following inductive transitions from one element of a chain to its nearest element in the lattice are used: (i) from s_q to s_{q+1}, (ii) from t_q to t_{q+1}, (iii) from s_q to s_{q-1}, (iv) from t_q to t_{q-1}, where q, q+1, q-1 are the cardinalities of enumerated subsets of objects and values: s_q, s_{q+1}, and s_{q-1}, $\subseteq G$; t_q, t_{q+1}, and $t_{q-1} \subseteq M$.

The transitions can be smooth and boundary. Under smooth transition, generating sets of values (objects) is performed with preserving a given property of them. These properties are, for example, “to be a test for a given class of objects”, “to be an irredundant set of values”, “not to be a test for a given class of objects”, and some others. A transition is said to be boundary if it changes a given property of sets of values (objects) into the opposite one.

For realizing the smooth inductive transitions, the following inductive reasoning rules are used: generalization rule, specification rule, and dual generalization and specification rules.

The generalization rule is used to get all the sets of objects $s_{q+1} = \{i_1, i_2, \ldots i_q, i_{q+1}\}$ from a set $s_q = \{i_1, i_2, \ldots i_q\}$ such that (s_q, $val(s_q)$) and (s_{q+1}, $val(s_{q+1})$) are tests for a given class of objects. The termination condition of generalization chain is: for all the extension s_{q+1} of s_q, (s_{q+1}, $val(s_{q+1})$) is not a test for a given class of objects.

The specification rule is used to get all the sets of values $t_{q+1} = \{m_1, m_2, \ldots, m_{q+1}\}$ from a set $t_q = \{m_1, m_2, \ldots, m_q\}$ such that t_q and t_{q+1} are irredundant sets of values and ($obj(t_q)$, t_q) and ($obj(t_{q+1})$, t_{q+1}) are not tests for a given class of objects. The termination condition for specification chain is: for all the extensions t_{q+1} of t_q, t_{q+1} is either a redundant set of values or a test for a given class of objects.

The dual generalization and specification rules relate to narrowing the collection of values and objects, respectively.

These rules realize the Joint Method of Agreement and Difference.

All inductive transitions take their interpretations in human mental acts. The extending of a set of objects with checking the satisfaction of a given condition is a typical method of inductive reasoning. In pattern recognition, the process of inferring hypotheses about the unknown values of some attributes is reduced to the maximal expansion of a collection of the known values of some attributes in such a way that none of the forbidden pairs of values would belong to this expansion. The contraction of a collection of values is used, for instance, in order to delete from it redundant or non-informative values. The contraction of a collection of objects is used, for instance, in order to isolate a certain cluster in a class of objects. Thus, we distinguish lemons in the citrus fruits.

The smooth transitions require the use of searching for admissible values (objects) for extending or narrowing the set of values (objects). Consider some methods for choosing objects admissible for extending s. Let S(test) be the partially ordered set of

elements $s = \{i_1, i_2, \ldots i_q\}$, $q = 1, 2, \ldots, nt - 1$ obtained as a result of generalizations and satisfying the following condition: (s, val(s)) is a test for a given class of positive objects, nt is the number of positive objects. Let *STGOOD* be the partially ordered set of elements s satisfying the condition: (s, val(s)) is a GMRT for a given class of positive objects.

Method 1. Suppose that S(test) and *STGOOD* are not empty and $s \in S$(test). Construct the set $V = \{\cup s', s \subseteq s', s' \in \{S(\text{test}) \cup STGOOD\}\}$. The set V is the union of all elements in S(test) and *STGOOD* containing s, hence, s is in the intersection of these elements. If we want an extension of s not to be included in any element of $\{S(\text{test}) \cup STGOOD\}$, we must use, for extending s, the objects not appearing simultaneously with s in V. The set of objects, candidates for extending s, is equal to CAND(s) = $nts \backslash V$, where $nts = \{\cup s, s \in S(\text{test})\}$.

An object $j^* \in$ CAND(s) is not admissible for extending s if at least for one object $i \in s$ the pair $\{i, j^*\}$ either does not correspond to a test or it corresponds to a good test (it belongs to *STGOOD*). Let Q be the set of forbidden pairs of objects for extending s: $Q = \{\{i, j\} \subseteq S(+)$: ($\{i, j\}$, val($\{i, j\}$) is not a test for a given class of positive objects $\}$. Then the set of admissible objects is *select*(s) = $\{i, i \in \text{CAND}(s)$: $(\forall j)$ $(j \in s)$, $\{i, j\} \notin \{STGOOD$ or $Q\}\}$. The set Q can be generated in the beginning of searching for all GMRTs for a given class of positive objects.

Method 2. In this method, the set CAND(s) is determined as follows. Let $s^* = \{s \cup j\}$ be an extension of s, where $j \notin s$. Then val(s^*) $\subseteq$ val(s). Hence the intersection of val(s) and val(j) must be not empty. The set CAND(s) = $\{j: j \in nts \backslash s$, val($j$) $\cap$ val(s) $\neq \emptyset\}$.

The knowledge acquired during the process of generalization (the sets Q, CAND(s), S(test), *STGOOD*) is used for pruning the search in the domain space.

The boundary inductive transitions are used to get: (1) all the sets t_q from a set t_{q-1} such that (obj(t_{q-1}), t_{q-1}) is not a test but (obj(t_q), t_q) is a test, for a given set of objects; (2) all the sets t_{q-1} from a set t_q such that (obj(t_q), t_q) is a test, but (obj(t_{q-1}), t_{q-1}) is not a test for a given set of objects; (3) all the sets s_{q-1} from a set s_q such that (s_q, val(s_q)) is not a test, but (s_{q-1}, val(s_{q-1})) is a test for a given set of objects; (4) all the sets of s_q from a set s_{q-1} such that (s_{q-1}, val(s_{q-1})) is a test, but (s_q,val(s_q)) is not a test for a given set of objects. The boundary inductive transitions realize the Method of Difference or Method of Concomitant Changes. For their implementation, we use the inductive diagnostic rule (IDR) and dual inductive diagnostic rule (DIDR). These rules require searching for essential values (IDRs) and essential objects (DIDRs).

All the boundary transitions are also interpreted as human reasoning operations. Transition 1 is used for distinguishing two diseases with similar symptoms. Transition 2 can be interpreted as including a certain class of objects into a more general one. For instance, squares can be named parallelograms, all whose sides are equal. In some intellectual psychological texts, a task is given to remove the "superfluous" (inappropriate) object from a certain group of objects (rose, butterfly, phlox, and dahlia) (transition 3). Transition 4 can be interpreted as the search for a refuting example.

Inductive reasoning rules generate implicative assertions or logical rules of the first kind, as shown in Table 4.

Table 4. Rules of the first kind obtained with the use of inductive reasoning rules

Inductive rules	Action	Inferring rules of the first kind
Generalization rule	Extending s (narrowing t)	Implications
Specification rule	Extending t (narrowing s)	Implications
Inductive diagnostic rule	Searching for essential values	Diagnostic rules, forbidden rules
Dual inductive diagnostic rule	Searching for essential objects	Compatibility rules (approximate implications)

During the lattice construction, the implicative assertions based on tests, are generated and used immediately. The knowledge acquired during the process of generalization (specialization) is used for pruning the search space (current context) with the use of deductive reasoning rules.

4 Conclusion

This work is an attempt to consider a large class of machine-learning tasks as a model of commonsense reasoning process based on using well-known deduction and induction logical rules. For this goal, we have chosen the task of inferring good classification tests for a given partitioning on a given set of objects because a lot of well-known machine-learning problems such as inferring functional, implicative, and associative dependencies from a dataset are reduced to this task.

5 References

1. Mill, J. S.: The system of logic. Russian Publishing Company “Book Affair”, Moscow (1900) (in Russian).
2. Piaget, J.: Genesis of the elementary logical structures. Classification and serializations. “EKSMO-Press“, Moscow (2002) (in Russian).
3. Finn, V.K.: Synthesis of cognitive procedures and induction problem. VINITI, NTI, Ser. 2(1, 2) (1999) (in Russian).
4. Zakrevskij, A.D.: A common logical approach to data mining and pattern recognition. In: Triantaphyllou, E. and Felici, G. (eds.). Data mining and knowledge discovery approaches based on rule induction techniques (pp. 1-43). Springer (2004).
5. Naidenova, X.: Reducing machine learning tasks to the approximation of a given classification on a given set of examples. In Proceedings of the 5-th National Conference at Artificial Intelligence (Vol. 1, pp. 275-279), Kazan, Tatarstan (1996) (in Russian).
6. Shreider, J.: Algebra of classification. VINITI, NTI, Series 2 (9), pp. 3-6 (1974) (in Russian).
7. Boldyrev, N. G.: Minimization of Boolean partial functions with a large number of “Don’t Care” conditions and the problem of feature extraction. Proceedings of International Symposium “Discrete Systems” (pp.101-109). Riga, Latvia (1974).
8. Ore, O.: Galois Connections. Transactions of the American Mathematical Society, 55(1), 493-513 (1944).
9. Kuznetsov, S. O.: Machine learning on the basis of Formal Concept Analysis. Automation and Remote Control, 62(10), pp. 1543-1564 (2001).

Finding Errors in New Object in Formal Contexts

Artem Revenko[1,2]*, Sergei O. Kuznetsov[2], and Bernhard Ganter[1]

[1] Technische Universität Dresden
Zellescher Weg 12-14, 01069 Dresden, Germany
[2] National Research University Higher School of Economics
Pokrovskiy bd. 11, 109028 Moscow, Russia
artem_viktorovich.revenko@mailbox.tu-dresden.de,
skuznetsov@hse.ru, bernhard.ganter@tu-dresden.de

Abstract. Classification of possible errors in formal contexts is given and the possibilities of exploring them are discussed. An approach for finding errors of some classes in formal contexts is introduced. This approach may be used in attempt to find errors in an object that is to be added to the context. The idea is based on finding those implications from an implication basis of the context, which are not respected by the object. It is noted that addressing such a problem directly may lead to an intractable solution. Alternative approach based on closing subsets of the intent of an object is considered in order to be able to find solution in polynomial time and deal with inconsistent combination of attributes. Examples are provided.

Keywords: formal context, implication, error exploration

1 Introduction

The work is motivated by the idea of building multi-user system based on Formal Concept Analysis methods. It would be different from QED project [QED] in the way that information should not be formalised as mathematical expressions and from Wikipedia in the way that information can somehow be inferenced by computer. In such a multi-user system error finding tools are absolutely necessary. In this work only errors in new objects (not yet added to the context) are considered. Throughout the text we assume that objects in the context are checked by an expert and correct. We attempt to find errors in new objects based on the information already in the context. This is actually the first step to building error finding tools.

2 Main Definitions

Let G and M be given sets. Let $I \subseteq G \times M$ be a binary relation between G and M. Triple $\mathbb{K} := (G, M, I)$ is called a *(formal) context*.
Set G is called a set of *objects*. Set M is called a set of *attributes*.

* We thank Sergei Obiedkov for discussion and useful remarks

Consider mappings $\varphi\colon 2^G \to 2^M$ and $\psi\colon 2^M \to 2^G$: $\varphi(A) := \{m \in M \mid gIm \text{ for all } g \in A\}$, $\psi(B) := \{g \in G \mid gIm \text{ for all } m \in B\}$. For any $A_1, A_2 \subseteq G$, $B_1, B_2 \subseteq M$ one has

1. $A_1 \subseteq A_2 \Rightarrow \varphi(A_2) \subseteq \varphi(A_1)$
2. $B_1 \subseteq B_2 \Rightarrow \psi(B_2) \subseteq \psi(B_1)$
3. $A_1 \subseteq \psi\varphi(A_1)$ and $B_1 \subseteq \varphi\psi(B_1)$

Mappings φ and ψ define a *Galois connection* between $(2^G, \subseteq)$ and $(2^M, \subseteq)$, i.e. $\varphi(A) \subseteq B \Leftrightarrow \psi(B) \subseteq A$. Usually, instead of φ and ψ a single notation $(\cdot)'$ is used. $(\cdot)'$ is sometimes called a *derivation operator*. For object $g \in G$ the set $g' = \{m \in M | gIm\}$ is called an *intent* of g and is denoted $\mathrm{int}(g)$. Similarly, for attribute $m \in M$ the set m' is called an *extent* of m and is denoted $\mathrm{ext}(m)$. Let $\overline{M} = \{\overline{m} \mid m \in M\}$ and $\overline{I} = \{(g, m) \mid g \in G, m \in M, (g, m) \notin I\}$. Triple $\overline{\mathbb{K}} := (G, \overline{M}, \overline{I})$ is called a *complementary (formal) context*.
Let $B \subseteq M, g \in G, \overline{B} := \{\overline{b} \mid b \in B\}$. $\overline{B} \subseteq \mathrm{int}(g)$ means that in $\mathbb{K} = (G, M, I)$ object g is not related to all attributes from B.

An *implication* of $\mathbb{K} := (G, M, I)$ is defined as a pair (A, B), written $A \to B$, where $A, B \subseteq M$. A is called a *premise*, B is called a *conclusion*. Implication $A \to B$ is *respected by a set of attributes* N if $A \nsubseteq N$ or $B \subseteq N$. Implication $A \to B$ holds (is valid) in $\mathbb{K}$ if $A' \subseteq (B \cup A)'$ or $A' \subseteq B'$, i.e. every object, that has all the attributes from A, also has all the attributes from B. The implications of $\mathbb{K}$ satisfy *Armstrong rules*:

$$\frac{}{X \to X} \quad , \quad \frac{X \to Y}{X \cup Z \to Y} \quad , \quad \frac{X \to Y, Y \cup Z \to W}{X \cup Z \to W}$$

A *support* of an implication is the set of object, whose intents respect this implication.

An *implication basis* of context $\mathbb{K}$ is defined as a set $\mathfrak{L}$ of implications of $\mathbb{K}$, from which any valid implication for $\mathbb{K}$ can be deduced by the Armstrong rules and none of the proper subsets of $\mathfrak{L}$ has this property.
A minimal implication basis is an implication basis minimal in the number of implications. A minimal implication basis was defined in [GD86] and is known as *canonical implication basis*. In paper [Gan84] the premises of implications from canonical base were characterized in terms of pseudo-intents. A subset of attributes $P \subseteq M$ is called *pseudo-intent*, if $P \neq P''$ and for every such pseudo-intent Q such that $Q \subset P$, one has $Q'' \subset P$. Canonical implication basis looks then as follows: $\{P \to (P'' \setminus P) \mid P \text{ - pseudo-intent}\}$.

3 Classification and Exploration of Errors

Every object in the context is described by the set of attributes, that are related to this object. In the ''real world'' there may exist dependencies between attributes. Consider possible cases in terms of implications:

1. Valid in ''real world'' dependency $A \to B$, $A, B \subseteq M$ is not respected by an object

2. Valid in ''real world'' dependency $A \to \overline{B},\ A, B \subseteq M$ is not respected by an object
3. Combination of two above cases, i.e. valid in ''real world'' dependency $A \to \overline{B} \wedge C,\ A, B, C \subseteq M$ is not respected by an object
4. Valid in ''real world'' dependency $A \to b \vee c,\ A \subseteq M, b, c \in M$ is not respected by an object
5. Valid in ''real world'' dependency $A \to \mathbf{F}$, where $A \subseteq M, \mathbf{F}$ is any logical formula not considered above, is not respected by an object (for example, $\mathbf{F} = a \vee (b \wedge \bar{c})$)

Unfortunately, it turns out that not all possible errors might be found using implications of a context. Namely, case 4 corresponds to reducible object in a context, while it is known that reducible objects change neither the lattice nor the implication basis of a context (definitions of reducible objects and lattices of contexts are not given in this paper for the sake of compactness, for definitions and further information see [GW99]).
Every implication $A \to B$ can be regarded as a conjunction of implications $A \to B_1$ and $A \to B_2,\ B_1 \cup B_2 = B$. Thereby, in Case 5 in $\mathbf{F}$ top level conjunctions can be dealt with easily. However, as in Case 4 we do not know how to reveal errors that have disjunctions in their conclusion.

Let $\mathcal{L}$ be a set of all implications valid for a context $\mathbb{K}$. From any implication basis any valid implication for the context should be deducible by definition. It means that if an object does not respect an implication from $\mathcal{L}$, then it should not respect an implication from implication basis of the context. Then an expert is asked whether this implication is valid. If he accepts this implication, then the object is an error. All the errors of the first class are caught using this approach.

4 An Example

The formal context on Fig. 1 shows the properties of convex quadrangles. The context is not full, i.e. not all possible convex quadrangles are considered, and some objects in the context are reducible (do not bring new information in an implication basis of the context). 7 attributes are considered. Attributes 'has equal legs' and 'has equal angles' require at least two angles/legs of a quadrangle to be equal. Some dependencies on attributes are obvious, for example, it is clear that if all angles are equal in a quadrangle then this quadrangle definitely has equal angles.
4 errors are presented on Fig 2. Errors are added to the context on Fig. 1 one at a time. One should treat an error as an object to be added to the context.
The context without errors on Fig. 1 is denoted $\mathbb{K}$, $(\cdot)'$ is the corresponding derivation operator.
The context of errors on Fig. 2 is denoted $\mathbb{K}_\mathrm{e}$, $(\cdot)^\mathrm{e}$ is the derivation operator for $\mathbb{K}_\mathrm{e}$.

Example 1. $\{\text{Error 1}\}^\mathrm{e}$ =$\{$has equal legs, at least 3 different angles, all legs equal$\}$
$\{\text{Error 1}\}^{\mathrm{e}\prime\prime}$ =$\{$all angles equal, all legs equal, at least 3 different angles, at least 3 different legs, has equal angles, has equal legs, has right angle$\}$

Convex quadrangles	has equal legs	has equal angles	has right angle	all legs equal	all angles equal	at least 3 different angles	at least 3 different legs
Square	×	×	×	×	×		
Rectangle	×	×	×		×		
Quadrangle						×	×
Rhombus	×	×		×			
Parallelogram	×	×					
Rectangular trapezium		×	×			×	×
Quadrangle with 2 equal legs and right angle	×		×			×	×
Isosceles trapezium	×	×				×	
Rectangular trapezium with 2 equal legs	×	×	×			×	×
Quadrangle with 2 equal angles		×				×	×
Quadrangle with 2 equal legs	×					×	×
Quadrangle with 2 equal legs and 2 equal angles	×	×				×	×

Fig. 1. Context of convex quadrangles $\mathbb{K}$

Errors	has equal legs	has equal angles	has right angle	all legs equal	all angles equal	at least 3 different angles	at least 3 different legs
Error1	×			×		×	
Error2	×		×	×	×		
Error3		×	×	×	×	×	×
Error4	×	×		×			×

Fig. 2. Context of errors $\mathbb{K}_e$

Canonical basis of the context on Fig. 1 looks as follows:

1. at least 3 different angles → at least 3 different legs
2. all angles equal → has equal angles, has equal legs, has right angle
3. all legs equal → has equal angles, has equal legs
4. has right angle, at least 3 different legs → at least 3 different angles
5. has equal angles, has equal legs, at least 3 different legs, all legs equal → has right angle,at least 3 different angles, all angles equal
6. has equal angles, has equal legs, at least 3 different legs, all angles equal, has right angle, at least 3 different angles → all legs equal
7. has right angle, has equal legs, all legs equal, has equal angles → all angles equal

Consider Error2.
$\{\text{Error 2}\}^{\text{e}}$ ={has equal legs, has right angle, all legs equal, all angles equal}
This object does not respect Implications 2 and 3. It is easy to see that both implications are valid in "real world". Thereby, an expert recognizes object as an error.

5 Improvements

Although this approach gives the needed result, there are some problems remaining. The problem of producing canonical basis with known algorithms is intractable. Recent theoretical results suggest that existing approaches for computing the stem base may not lead to algorithms with better worst-case complexity [DS11], [BK10]. One can use other bases (for example, advances were achieved in computing proper premises [RDB11]), but known algorithms are still too costly and non minimal bases do not guarantee to ask an expert minimal yet sufficient number of questions.
Since we are only interested in implication corresponding to an object, it is not necessary to compute a whole implication basis. Only the closure of object's intent may be considered. From monotonicity of the closure operator of the context it follows that we do not lose any attributes that are erroneously not related to an object. Nevertheless, the following case is possible. Let a set $H \subseteq M$ be the intent of an object such that $\nexists g \in G : H \subseteq g'$. In this case $H'' = M$ and the implication $H \to H'' \setminus H$ has empty support. This is the case if an object is an error of the second class, because in its intent impossible in "real world" combination of attributes is contained. Although it is not the best solution, but we can ask an expert if the combination of attributes in object's intent is consistent. In such a question we use information already input in the context, but an expert should consider all possible combinations of attributes to be excessive. Further on this question is not sufficient to reveal an error of the first class.
A better idea would be to investigate the subset of object's intent. Even if $H'' = M$, we are still able to reveal an error of the first class, because we examine all possible dependencies on the subset of object's intent that were satisfied by intents in the context. Searching through all the subsets of object's intent leads

to exponential time solution. Since we are only interested in such subsets that are contained in at least one intent in the context, we may consider only the intersections of object's intent with intents in the context. This allows us to find errors of the first class in polynomial time.
But we can do even better and replace the question about the consistency of set of attributes in object's intent with an implication. This idea is better because in case of implication an expert is explicitly shown an attribute that breaks the dependencies satisfied in the context. For this purpose we should consider complementary context. If we investigate the closures of the subsets of initial object's intent in the context $\mathbb{K}_\cup := \{G, M \cup \overline{M}, I \cup \overline{I}\}$, then we are also able to find errors of the second class in the very same manner, that we discussed before for the first class. Thus we are able to find all the errors of first three classes.

5.1 Pseudocode

Below is presented the pseudocode of the method described above.

```
inspect_with_negations(𝕂:=(G, M, I), H⊆M)
1. Candidates = {object′∩H | object∈G}
2. Candidates = {candidate∈Candidates |
                 ∄c∈Candidates: candidate⊆c}
3. Result = ∅
4. for candidate in Candidates:
    5. if candidate^{∪′∪} != candidate:
        6. Result.add(candidate → candidate^{∪′∪}\(candidate∪H))
7. return Result
```

`H` is the intent of an object to be added to the context $\mathbb{K}$. In the first line we compute the set of all subsets that could produce desired implication. Since closure operator is monotone we may consider only maximal elements of `Candidates`. In the second line we discard all the non-maximal elements. In line 4 - 6 we check if closure differs from the intersection, i.e. there are attributes in conclusion of future implication. Here $(\cdot)^{\prime^\cup}$ denotes derivation operator of the context $\mathbb{K}_\cup$. There is no need to check if `candidate`$''$ is contained in `H`, because all the attributes from `H`$\setminus$`candidate` are not contained in the intent of at least one object, from which this `candidate` was generated in the first line.
It is possible to provide further optimisation. In the first line we can stop generating `Candidates` after all the maximal subsets satisfying condition were found.

6 Results

FCA package for Python written by Nikita Romashkin was used for implementation [Rom].
The name `inspect_dg` is used to denote the function implementing the method described in Section 3.

Inspecting Error1:

`inspect\_dg`
at least 3 different angles → at least 3 different legs
all legs equal → has equal angles, has equal legs
`inspect\_with\_negation`
has equal legs, at least 3 different angles → at least 3 different legs, all legs equal
has equal legs, all legs equal → has equal angles, at least 3 different angles

Both implications in the result of v without overlined attributes in conclusions are deducible from the two implications in the result of `inspect_dg`. The two implications in the result of `inspect_dg` with the intent of Error1 added in the premise are deducible from the implications in the result of `inspect_with_negation`. Considering a particular object and corresponding implications we can always add object's intent to the premise(s), because attributes from its intent are always related to the object.
Errors of the second class are not caught using canonical base. As described above such errors correspond to inconsistent in the context combination of attributes. Nevertheless, `inspect_with_negation` catches such errors.

Inspecting Error2:

`inspect\_dg`
all angles equal → has equal angles, has equal legs, has right angle
all legs equal → has equal angles, has equal legs
`inspect\_with\_negation`
has right angle, has equal legs, all legs equal, all angles equal → has equal angles

In this example we are able to ask even less number of questions to the expert using `inspect_with_negation` as with `inspect_dg`. This is the result of finding implications generated by maximal subsets of object's intent. Again, adding object's intent to the premises of implications in the result of `inspect_dg` makes both groups mutually deducible.

Inspecting Error3:

`inspect\_dg`
all angles equal → has equal angles, has equal legs, has right angle
all legs equal → has equal angles, has equal legs
`inspect\_with\_negation`
has equal angles, has right angle, at least 3 different legs, at least 3 different angles → all angles equal, all legs equal
has equal angles, has right angle, all legs equal, all angles equal → has equal legs, at least 3 different angles, at least 3 different legs

The case of Error3 is more or less the same, as the case of Error1. The two groups of implications are mutually deducible under the same conditions as before.

Inspecting Error4:

`inspect\_dg`
has equal angles, has equal legs, at least 3 different legs, all legs equal → has right angle, at least 3 different angles, all angles equal

`inspect\_with\_negation`
has equal angles, has equal legs, all legs equal → $\overline{\text{at least 3 different legs}}$
has equal angles, has equal legs, at least 3 different legs → $\overline{\text{all legs equal}}$

Error4 is a very special case when corresponding implication from canonical basis has empty support. In this case even if implication from the result of `inspect_dg` is rejected by an expert object may still be an error. This implication is in fact excessive, because the premise is not contained in any intent in the context and all attributes, that are not in premise, are in conclusion. Using `inspect_with_negation` we are able to ask an expert more sensible questions. Unfortunately, groups of implications are not deducible from each other in this example.

7 Conclusion

An algorithm for finding errors in new objects of the context was proposed. As opposed to finding not respected implications in an implication basis proposed algorithm finishes in polynomial time. Moreover, in case of inconsistent combination of attributes in object's intent it is possible to state more sensible questions. In some cases the number of produced questions to an expert is less than the number of not respected implications in the canonical basis of the context.

References

[BK10] M. Babin and S. O. Kuznetsov. Recognizing pseudo-intent is conp-complete. *Proc. 7th International Conference on Concept Lattices and Their Applications, University of Sevilla*, pages 294–301, 2010.

[DS11] Felix Distel and Barış Sertkaya. On the complexity of enumerating pseudo-intents. *Discrete Applied Mathematics*, 159(6):450–466, 2011.

[Gan84] B. Ganter. Two basic algorithms in concept analysis. *Preprint-Nr. 831*, 1984.

[GD86] J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Math. Sci. Hum*, 24(95):5–18, 1986.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis : Mathematical Foundations*. Springer, 1999.

[QED] The qed project. http://mizar.org/qed/.

[RDB11] Uwe Ryssel, Felix Distel, and Daniel Borchmann. Fast computation of proper premises. In Amedeo Napoli and Vilem Vychodil, editors, *International Conference on Concept Lattices and Their Applications*, pages 101–113. INRIA Nancy – Grand Est and LORIA, 2011.

[Rom] Nikita Romashkin. Python package for formal concept analysis. https://github.com/jupp/fca.

Finding minimal rare itemsets in a depth-first manner

Laszlo Szathmary[1], Petko Valtchev[2], Amedeo Napoli[3], and Robert Godin[2]

[1] University of Debrecen, Faculty of Informatics, Department of IT,
H-4010 Debrecen, Pf. 12, Hungary
Szathmary.L@gmail.com
[2] Dépt. d'Informatique UQAM, C.P. 8888,
Succ. Centre-Ville, Montréal H3C 3P8, Canada
{valtchev.petko, godin.robert}@uqam.ca
[3] LORIA UMR 7503, B.P. 239, 54506 Vandœuvre-lès-Nancy Cedex, France
napoli@loria.fr

Abstract. Rare itemsets are an important sort of patterns that have a wide range of practical applications. Although mining rare patterns poses specific algorithmic problems, it is yet insufficiently studied. In a previous work, we proposed a levelwise approach for rare itemset mining. Here, we examine the benefits of depth-first methods for that task as such methods are known to outperform the levelwise ones in many practical cases.

1 Introduction

Pattern mining is a basic data mining task whose aim is to uncover the hidden regularities in a set of data [1]. As a simplifying hypothesis, the overwhelming majority of pattern miners chose to look exclusively on item combinations that are sufficiently frequent, i.e., observed in a large enough proportion of the transactions. Yet such a hypothesis fails to reflect the entire variety of situations in data mining practice [2]. In some specific situations, frequency may be the exact opposite of pattern interestingness. The reason behind is that in these cases, the most typical item combinations from the data correspond to widely-known and well-understood phenomena. In contrast, less frequently occurring patterns may point to unknown or poorly studied aspects of the underlying domain [2].

In a previous paper [3], we proposed a bottom-up, levelwise approach that traverses the frequent zone of the search space. In this paper we are looking for a more efficient manner for traversing the frequent part of the Boolean lattice, using a depth-first method. Indeed, depth-first frequent pattern miners have been shown to outperform breadth-first ones on a number of datasets.

2 Basic Concepts

Consider the following 6×5 sample dataset: $\mathcal{D} = \{(1,\ ABCDE),\ (2,\ BCD),\ (3,\ ABC),\ (4,\ ABE),\ (5,\ AE),\ (6,\ DE)\}$. Throughout the paper, we will refer to this example as **"dataset $\mathcal{D}$"**.

Consider a set of *objects* or *transactions* $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$, a set of *attributes* or *items* $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, and a relation $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$. A set of items is called an *itemset*. Each transaction has a unique identifier (*tid*), and a set of transactions is called a *tidset*. The tidset of all transactions sharing a given itemset X is its *image*, denoted by $t(X)$. The *length* of an itemset X is $|X|$, whereas an itemset of length i is called an i-itemset. The (absolute) *support* of an itemset X, denoted by $supp(X)$, is the size of its image, i.e. $supp(X) = |t(X)|$.

The lattice is separated into two segments or zones through a user-provided "minimum support" threshold, denoted by *min_supp*. Thus, given an itemset X, if $supp(X) \geq min\_supp$, then it is called *frequent*, otherwise it is called *rare* (or *infrequent*). In the lattice in Figure 1, the two zones corresponding to a support threshold of 2 are separated by a solid line. The rare itemset family and the corresponding lattice zone is the target structure of our study.

Definition 1. X subsumes Z, *iff* $X \supset Z$ *and* $supp(X) = supp(Z)$ [4].

Definition 2. *An itemset Z is* generator *if it has no proper subset with the same support.*

Property 1. Given $X \subseteq \mathcal{A}$, if X is a generator, then $\forall Y \subseteq X$, Y is a generator, whereas if X is not a generator, $\forall Z \supseteq X$, Z is not a generator [5].

Proposition 1. *An itemset X is a generator iff* $supp(X) \neq min_{i \in X}(supp(X \setminus \{i\}))$ [6].

Each of the frequent and rare zones is delimited by two subsets, the maximal elements and the minimal ones, respectively. The above intuitive ideas are formalized in the notion of a border introduced by Mannila and Toivonen in [7]. According to their definition, the maximal frequent itemsets constitute the *positive border* of the frequent zone[1] whereas the minimal rare itemsets form the *negative border* of the same zone.

Definition 3. *An itemset is a* maximal frequent itemset *(MFI) if it is frequent but all its proper supersets are rare.*

Definition 4. *An itemset is a* minimal rare itemset *(mRI) if it is rare but all its proper subsets are frequent.*

The levelwise search yields as a by-product all mRIs [7]. Hence we prefer a different optimization strategy that still yields mRIs while traversing only a subset of the frequent zone of the Boolean lattice. It exploits the minimal generator status of the mRIs. By Property 1, frequent generators (FGs) can be traversed in a levelwise manner while yielding their negative border as a by-product. It is enough to observe that mRIs are in fact generators:

Proposition 2. *All minimal rare itemsets are generators* [3].

[1] The frequent zone contains the set of frequent itemsets.

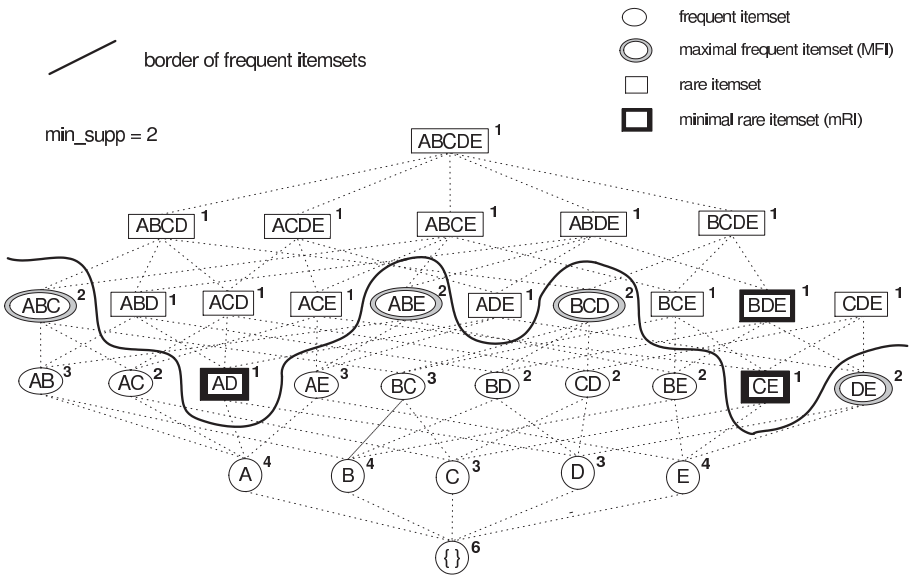


Fig. 1. The powerset lattice of dataset $\mathcal{D}$.

Finding Minimal Rare Itemsets in a Levelwise Manner

As pointed out by Mannila and Toivonen [7], the easiest way to reach the negative border of the frequent itemset zone, i.e., the mRIs, is to use a levelwise algorithm such as *Apriori*. Indeed, albeit a frequent itemset miner, *Apriori* yields the mRIs as a by-product.

Apriori-Rare [3] is a slightly modified version of *Apriori* that retains the mRIs. Thus, whenever an i-long candidate survives the frequent $i-1$ subset test, but proves to be rare, it is kept as an mRI.

MRG-Exp [3] produces the same output as *Apriori-Rare* but in a more efficient way. Following Proposition 2, *MRG-Exp* avoids exploring all frequent itemsets: instead, it looks after frequent generators only. In this case mRIs, which are rare generators as well, can be filtered among the negative border of the frequent generators. The output of *MRG-Exp* is identical to the output of *Apriori-Rare*, i.e. both algorithms find the set of mRIs.

3 Finding Minimal Rare Itemsets in a Depth-First Manner

Eclat [8] was the first FI-miner to combine the vertical encoding with a depth-first traversal of a tree structure, called IT-tree, whose nodes are $X \times t(X)$ pairs. *Eclat* traverses the IT-tree in a depth-first manner in a pre-order way, from left-to-right [8] (see Figure 2).

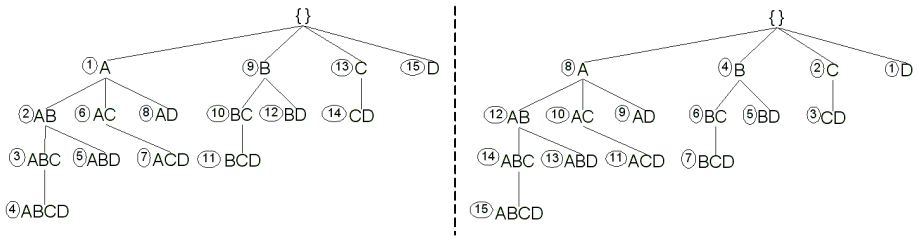


Fig. 2. Left: pre-order traversal with *Eclat*; **Right:** reverse pre-order traversal with *Eclat*. The direction of traversal is indicated in circles

3.1 Talky-G

Talky-G [9] is a vertical FG-miner following a depth-first traversal of the IT-tree and a right-to-left order on sibling nodes. *Talky-G* applies an inclusion-compatible traversal: it goes down the IT-tree while listing sibling nodes from right-to-left and not the other way round as in *Eclat*.

The authors of [10] explored that order for mining calling it *reverse pre-order*. They observed that for any itemset X its subsets appear in the IT-tree in nodes that lay either higher on the same branch as $(X, t(X))$ or on branches to the right of it. Hence, depth-first processing of the branches from right-to-left would perfectly match set inclusion, i.e., all subsets of X are met before X itself. While the algorithm in [10] extracts the so-called non-derivable itemsets, *Talky-G* uses this traversal to find the set of frequent generators. See Figure 2 for a comparison of *Eclat* and its "reversed" version.

3.2 Walky-G

Since *Walky-G* is an extension of *Talky-G*, we also present the latter algorithm at the same time. *Walky-G*, in addition to *Talky-G*, retains rare itemsets and checks them for minimality.

Hash structure. In *Walky-G* a hash structure is used for storing the already found frequent generators. This hash, called $fgMap$, is a simple dictionary with key/value pairs, where the key is an itemset (a frequent generator) and the value is the itemset's support.[2] The usefulness of this hash is twofold. First, it allows a quick look-up of the proper subsets of an itemset with the same support, thus the generator status of a frequent itemset can be tested easily (see Proposition 1). Second, this hash is also used to look-up the proper subsets of a minimal rare candidate. This way rare but non-minimal itemsets can be detected efficiently.

Pseudo code. Algorithm 1 provides the main block of *Walky-G*. First, the IT-tree is initialized, which involves the creation of the root node, representing

[2] In our implementation we used the `java.util.HashMap` class for $fgMap$.

Algorithm 1 (main block of Walky-G):

```
 1) // for quick look-up of (1) proper subsets with the same support
 2) // and (2) one-size smaller subsets:
 3) fgMap ← ∅   // key: itemset (frequent generator); value: support
 4)
 5) root.itemset ← ∅   // root is an IT-node whose itemset is empty
 6) // the empty set is included in every transaction:
 7) root.tidset ← {all transaction IDs}
 8) fgMap.put(∅, |O|)   // the empty set is an FG with support 100%
 9) loop over the vertical representation of the dataset (attr) {
10)      if (min_supp ≤ attr.supp < |O|) {
11)           // |O| is the total number of objects in the dataset
12)           root.addChild(attr)   // attr is frequent and generator
13)      }
14)      if (0 < attr.supp < min_supp) {
15)           saveMri(attr)   // attr is a minimal rare itemset
16)      }
17) }
18) loop over the children of root from right-to-left (child) {
19)      saveFg(child)   // the direct children of root are FGs
20)      extend(child)   // discover the subtree below child
21) }
```

the empty set (of 100% support, by construction). *Walky-G* then transforms the layout of the dataset in vertical format, and inserts below the root node all 1-long frequent itemsets. Such a set is an FG whenever its support is less than 100%. Rare attributes (whose support is less than min_supp) are minimal rare itemsets since all their subsets (in this case, the empty set) are frequent. Rare attributes with support 0 are not considered.

The `saveMri` procedure processes the given minimal rare itemset by storing it in a database, by printing it to the standard output, etc. At this point, the dataset is no more needed since larger itemsets can be obtained as unions of smaller ones while for the images intersection must be used.

The `addChild` procedure inserts an IT-node under a node. The `saveFg` procedure stores a given FG with its support value in the hash structure $fgMap$.

In the core processing, the `extend` procedure (see Algorithm 2) is called recursively for each child of the root in a right-to-left order. At the end, the IT-tree contains all FGs. Rare itemsets are verified during the construction of the IT-tree and minimal rare itemsets are retained. The `extend` procedure discovers all FGs in the subtree of a node. First, new FGs are tentatively generated from the right siblings of the current node. Then, certified FGs are added below the current node and later on extended recursively in a right-to-left order.

The `getNextGenerator` function (see Algorithm 3) takes two nodes and returns a new FG, or "null" if no FG can be produced from the input nodes. In addition, this method tests rare itemsets and retains the minimal ones. First, a candidate node is created by taking the union of both itemsets and the intersection of their respective images. The input nodes are thus the candidate's

Algorithm 2 ("extend" procedure):

Method: extend an IT-node recursively (discover FGs in its subtree)
Input: an IT-node (*curr*)

```
1) loop over the right siblings of curr from left-to-right (other) {
2)     generator ← getNextGenerator(curr, other)
3)     if (generator ≠ null) then curr.addChild(generator)
4) }
5) loop over the children of curr from right-to-left (child) {
6)     saveFg(child)   // child is a frequent generator
7)     extend(child)   // discover the subtree below child
8) }
```

parents. Then, the candidate undergoes a frequency test (test 1). If the test fails then the candidate is rare. In this case, the minimality of the rare itemset *cand* is tested. If all its one-size smaller subsets are present in $fgMap$ then *cand* is a minimal rare generator since all its subsets are FGs (see Property 1). From Proposition 2 it follows that an mRG is an mRI too, thus *cand* is processed by the `saveMri` procedure. If the frequency test was successful, the candidate is compared to its parents (test 2): if its tidset is equivalent to a parent tidset, then the candidate cannot be a generator. Even with both outcomes positive, an itemset may still not be a generator as a subsumed subset may lay elsewhere in the IT-tree. Due to the traversal strategy in *Walky-G*, all generator subsets of the current candidate are already detected and the algorithm has stored them in $fgMap$ (see the `saveFg` procedure). Thus, the ultimate test (test 3) checks whether the candidate has a proper subset with the same support in $fgMap$. A positive outcome disqualifies the candidate.

This last test (test 3) is done in Algorithm 4. First, one-size smaller subsets of *cand* are collected in a list. The two parents of *cand* can be excluded since *cand* was already compared to them in test 2 in Algorithm 3. If the support value of one of these subsets is equal to the support of *cand*, then *cand* cannot be a generator. Note that when the one-size smaller subsets are looked up in $fgMap$, it can be possible that a subset is missing from the hash. It means that the missing subset was tested before and turned out to subsume an FG, thus the subset was not added to $fgMap$. In this case *cand* has a non-FG subset, thus *cand* cannot be a generator either (by Property 1).

Candidates surviving the final test in Algorithm 3 are declared FG and added to the IT-tree. An unsuccessful candidate X is discarded which ultimately prevents any itemset Y having X as a prefix to be generated as candidate and hence substantially reduces the overall search space. When the algorithm stops, all frequent generators (and *only* frequent generators) are inserted in the IT-tree *and* in the $fgMap$ structure. Furthermore, upon the termination of the algorithm, all minimal rare itemsets have been found. For a running example, see Figure 3.

Algorithm 3 ("getNextGenerator" function):

Method: create a new frequent generator *and* filter minimal rare itemsets
Input: two IT-nodes (*curr* and *other*)
Output: a frequent generator or null

```
 1) cand.itemset ← curr.itemset ∪ other.itemset
 2) cand.tidset ← curr.tidset ∩ other.tidset
 3) if (cardinality(cand.tidset) < min_supp)   // test 1: frequent?
 4) {   // now cand is an mRI candidate; let us test its minimality:
 5)     if (all one-size smaller subsets of cand are in fgMap) {
 6)         saveMri(cand)   // cand is an mRI, save it
 7)     }
 8)     return null   // not frequent
 9) }
10) // else, if it is frequent; test 2:
11) if ((cand.tidset = curr.tidset) or (cand.tidset = other.tidset)) {
12)     return null   // not a generator
13) }
14) // else, if it is a potential frequent generator; test 3:
15) if (candSubsumesAnFg(cand)) {
16)     return null   // not a generator
17) }
18) // if cand passed all the tests then cand is a frequent generator
19) return cand
```

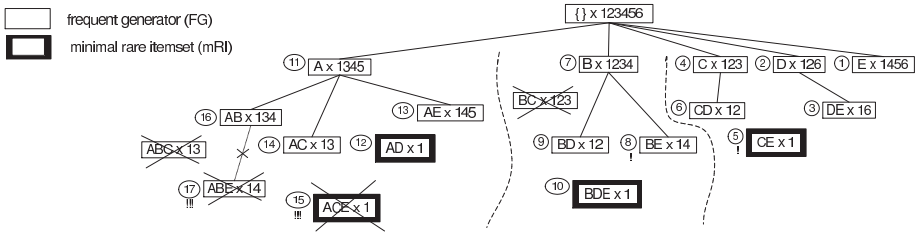


Fig. 3. The IT-tree built during the execution of *Walky-G* on dataset $\mathcal{D}$ with $min\_supp = 2$ (33%). Notice the two special cases: ACE is not an mRI because of CE; ABE is not an FG because of BE.

4 Conclusion

We presented an approach for rare itemset mining from a dataset that splits the problem into two tasks. Our new algorithm, *Walky-G*, limits the traversal of the frequent zone to frequent generators *only*. Our approach breaks with the dominant levelwise algorithmic schema since the traversal is achieved through a depth-first strategy.

Algorithm 4 ("candSubsumesAnFg" function):

Method: verify if *cand* subsumes an already found FG
Input: an IT-node (*cand*)

```
1) subsets ← {one-size smaller subsets of cand minus the two parents}
2) loop over the elements of subsets (ss) {
3)     if (ss is stored in fgMap) {
4)         stored_support ← fgMap.get(ss)   // get the support of ss
5)         if (stored_support = cand.support) {
6)             return true   // case 1: cand subsumes an FG
7)         }
8)     }
9)     else   // if ss is not present in fgMap
10)    {   // case 2: cand has a non-FG subset ⇒ cand is not an FG either
11)        return true
12)    }
13) }
14) return false   // if we get here then cand is an FG
```

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in knowledge discovery and data mining. American Association for Artificial Intelligence (1996) 307–328
2. Weiss, G.: Mining with rarity: a unifying framework. SIGKDD Explor. Newsl. **6**(1) (2004) 7–19
3. Szathmary, L., Napoli, A., Valtchev, P.: Towards Rare Itemset Mining. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '07). Volume 1., Patras, Greece (Oct 2007) 305–312
4. Zaki, M.J., Hsiao, C.J.: CHARM: An Efficient Algorithm for Closed Itemset Mining. In: SIAM International Conference on Data Mining (SDM' 02). (Apr 2002) 33–43
5. Kryszkiewicz, M.: Concise Representations of Association Rules. In: Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery. (2002) 92–109
6. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining Frequent Patterns with Counting Inference. SIGKDD Explor. Newsl. **2**(2) (2000) 66–75
7. Mannila, H., Toivonen, H.: Levelwise Search and Borders of Theories in Knowledge Discovery. Data Mining and Knowledge Discovery **1**(3) (1997) 241–258
8. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New Algorithms for Fast Discovery of Association Rules. In: Proceedings of the 3rd International Conference on Knowledge Discovery in Databases. (August 1997) 283–286
9. Szathmary, L., Valtchev, P., Napoli, A., Godin, R.: Efficient Vertical Mining of Frequent Closures and Generators. In: Proc. of the 8th Intl. Symposium on Intelligent Data Analysis (IDA '09). Volume 5772 of LNCS., Lyon, France, Springer (2009) 393–404
10. Calders, T., Goethals, B.: Depth-first non-derivable itemset mining. In: Proceedings of the SIAM International Conference on Data Mining (SDM '05), Newport Beach, USA. (Apr 2005)

A System for Knowledge Discovery in Big Dynamical Text Collections

Sergei O. Kuznetsov, Alexey A. Neznanov, Jonas Poelmans

National Research University Higher School of Economics,
Myasnitskaya 20, 101000, Moscow, Russian Federation
SKuznetsov@hse.ru, ANeznanov@hse.ru,
Jonas.Poelmans@econ.kuleuven.be

Abstract. Software system Cordiet-FCA is presented, which is designed for knowledge discovery in big dynamic data collections, including texts in natural language. Cordiet-FCA allows one to compose ontology-controlled queries and outputs concept lattice, implication bases, association rules, and other useful concept-based artifacts. Efficient algorithms for data preprocessing, text processing, and visualization of results are discussed. Examples of applying the system to problems of medical diagnostics, criminal investigations are considered.

Keywords: Formal Concept Analysis, Data Mining, Natural Language Processing, Software Tool, Visualization

1 Introduction

In this paper we introduce the software system Cordiet-FCA for data mining and knowledge discovery based on the Cordiet-DMS (Data Mining System) platform and used primarily the Formal Concept Analysis (FCA) [1] as theoretical basis. FCA emerged in the 1980's from attempts to restructure lattice theory in order to promote better communication between lattice theorists and potential users of lattice theory. Since its early years, Formal Concept Analysis has developed into a research field in its own right with a thriving theoretical community and a rapidly expanding range of applications in information and knowledge processing including visualization, data analysis (mining) and knowledge management.

The system was designed especially for unstructured data analysis. In case studies we applied Cordiet-FCA to the analysis of publications on FCA. The real-life datasets include criminal data (for example, chat conversations of pedophiles) and, in nearest future, medical and emergency rescue data.

2 Methodology

Software package Cordiet-DMS is a universal extendible software platform intended to build data mining and knowledge discovery tools for various application fields. This platform inspired by CORDIET methodology (abbreviation of Concept Relation Discovery and Innovation Enabling Technology) [2], developed by J. Poelmans in K.U. Leuven and P. Elzinga in Amsterdam-Amstelland police. The methodology allows one to obtain new knowledge from the data in iterative ontology-controlled process. The package is based on modern methods and algorithms of data analysis, technologies for manipulating big data collections, data visualization, reporting, and interactive processing techniques. There are four base principles:

1. Iterative process of data analysis using ontology-controlled queries and interactive artifacts (such as concept lattice, etc.).
2. Separation of processes of data querying (from various data sources) and data analyzing (of locally saved immutable snapshots).
3. Dividing data processing into four stages: access to external data sources and loading data to local storage; access to the local storage and generating snapshots; access to one or many snapshots and building basic analysis artifacts; access to the artifact and analyzing derivative artifacts.
4. Expendability on three levels: customizing settings of data access components, query builders, solvers and visualizers; writing scripts (macros); developing components (add-ins).

3 Current software properties

At this moment we introduce the version 0.9 of Cordiet-FCA in form of local Windows application. This version uses local XML-storage and integrated research environment with snapshot profiles editor, query builder, ontology editor, and a set of solvers and visualizers. The main solvers can produce concept lattice, sublattices, association rules, and implications, calculate stability indexes, similarity measures for contexts and concepts, etc.

We use Microsoft and Embarcadero programing environments and different programing languages (C++, C#, Delphi, Python and others). For scripting we use Delphi Web Script [3]. Also we are developing a distributed version based on Web-services.

3.1 Text processing

Cordiet-FCA has a query language for transforming data snapshot into basic analysis artifacts. The main artifact for FCA methods is a formal context.

The language describes so called rules and consists from four main rules types:

- Simple rule generates one attribute from structured fields of snapshot.
- Scaling rule generates several attributes from structured fields based on nominal or ordinal scale.

– Text mining rule generates one attribute from unstructured text fields.
– Multivalued rule generates one or many attributes from multivalued field (array).

Also we have temporal rules (for manipulating with date and time) and compound rules (for merging all types of rules into one). As usual we don't need to write a query from scratch. We can select some entities in the ontology editor and automatically generate a query. Text mining rule can use terms (set of synonymous) and term-clusters (set of terms) from ontology entities.

Cordiet-FCA uses Lucene [4] to index the content of the unstructured text fields in the snapshots using the description of the term attributes in the ontology editor. The resulting index is later used to quickly validate whether the text mining or compound rule return true or not. In fig. 1 we show how system visualizes the profile-controlled description of snapshots records (Report Viewer) and query builder (the list of rules and Rule Editor).

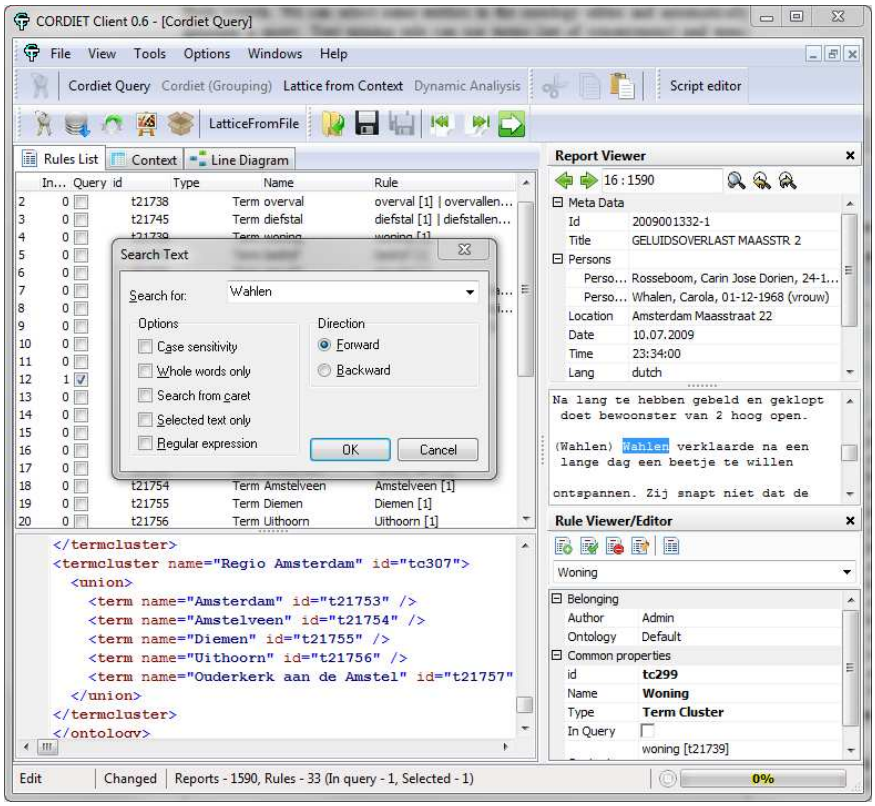


Fig. 1. Opened base of police reports and query builder

3.2 Concept lattice browser

The main mode of user interaction in Cordiet-FCA is interactive work in the concept lattice browser. The lattice can be used to browse the collection of objects with binary attributes given as a result of query to snapshot (with structured and text attributes). The user can select and deselect objects and attributes and the lattice diagram is modified accordingly. The user can click on a concept. The screen shows in a separate window the names of the objects in the extent and the names of the attributes in the intent. Names of objects and attributes are linked with initial snapshot records and fields. If the user clicks on the name of an object, the content of the object is shown in a separate window according to snapshot profile. If the user clicks on the name of an attribute, its content is also shown in a separate window.

Fig. 2 demonstrates the browser (building sublattice). The multidocument interface allows us to inspect several lattices and moreover the system remembers all links between derivative artifacts.

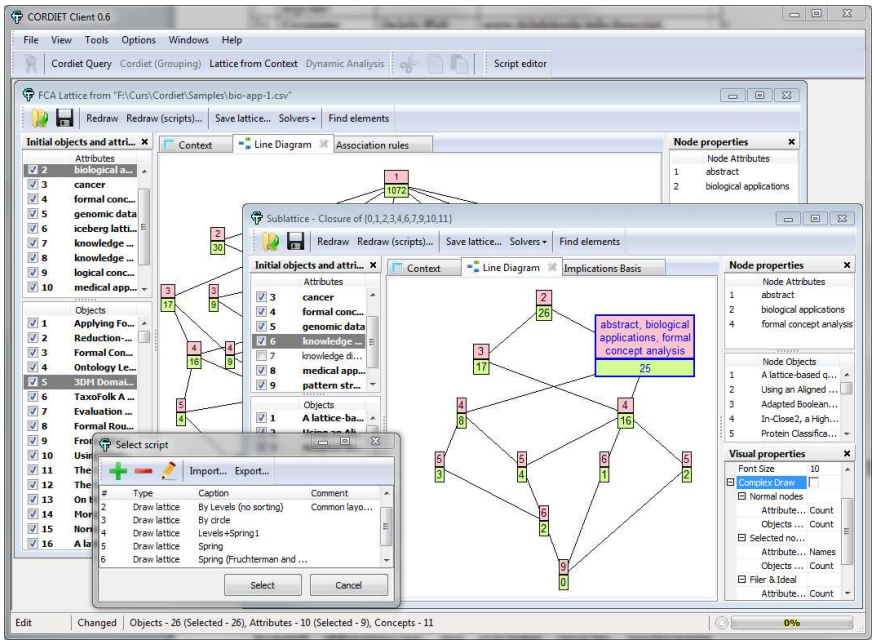


Fig. 2. Concept lattice browser

The user can customize the lattice browsing settings. The user can specify whether the nodes corresponding to concepts show the numbers of all (or only the new) objects and all (or only the new) attributes in extent and intent respectively, or the names of all (or only the new) objects and all (or only the new) attributes. Separate settings can be specified for the selected concept, the concepts in the order filter and the remainder of the lattice. If the user presses shift and at the same time selects a concept,

the order filter generated by this concept is shown. The colors of concepts and edges can be customized also.

A right click on the name of an attribute shows the user several options: the user can choose to build a sublattice containing only objects having the selected attribute, to build a sublattice containing only objects which do not have the selected attribute, or to find the concept in which the attribute first occurs starting from the supremum of the lattice.

3.3 Validation and applications

Main solvers of the system were validated on the classical test sets (from Frequent Itemset Mining Dataset Repository and UCI Repository). Because of constant improving of basic algorithms and data structures we don't have a good comparable set of benchmarks now.

We used Cordiet-FCA in the research work of the Laboratory of Intelligent Systems and Structural Analysis in NRU HSE and in some applied tasks connected with medical informatics, crime investigations, etc. Fig. 3 demonstrates an example of analyses of a pedophile behavior (it is based on information from chat conversations in Internet).

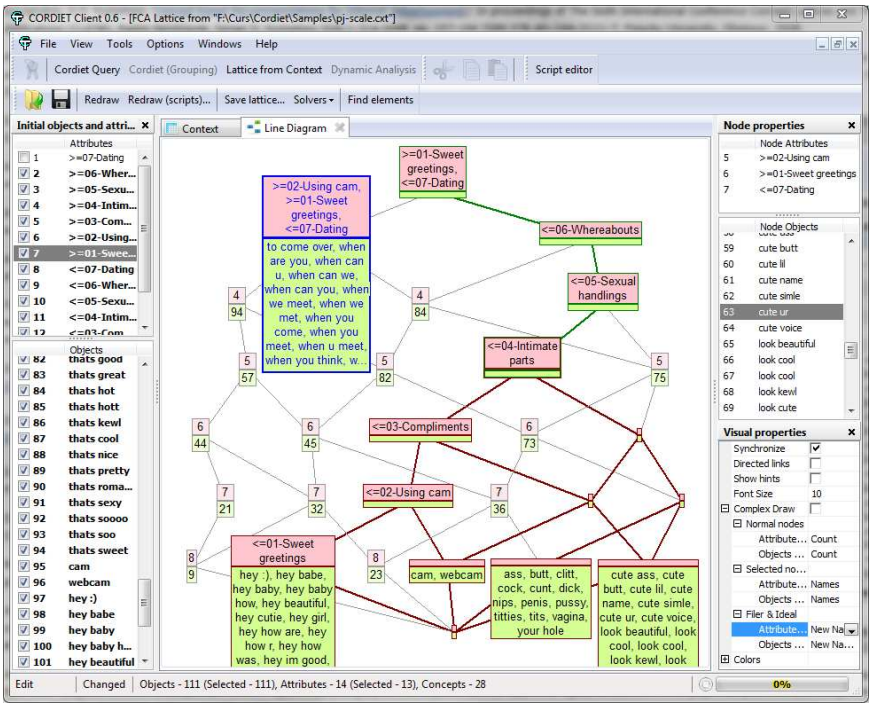


Fig. 3. Sample of concepts exploration (filter, ideal and selection of attributes)

4 Comparison with existing well-known FCA software

Comparing Cordiet-FCA with big analytic software like IBM i2 Analyst's Notebook or QSR NVivo shows that the latter do not have a normal set of FCA tools and have a completely different methodology of data analysis. We also compare basic functionality of the system with well-known tools for building and visualizing FCA artifacts (table 1).

Table 1. Some well-known FCA software tools

Program title	Author	Web-site
Concept Explorer	S.A.Evtuchenko	conexp.sourceforge.net
FcaStone	U. Priss et al	fcastone.sourceforge.net
Conflexplore	P.Borza, O.Sabou	code.google.com/p/openfca
Galicia	P.Valtchev et al	www.iro.umontreal.ca/~galicia
ToscanaJ	University of Queensland, Technical University of Darmstadt	toscanaj.sourceforge.net

All of the tools from Table 1 have unique features. For example, Concept Explorer has interesting modes of visualization of a lattice and good default layout, Galicia introduces the generic MultiFCA approach to deal with a set of contexts, ToscanaJ can visualize nested lattices and involves an editor of conceptual schemas on relational databases, FcaStone was primarily intended for file format conversion and other low level operations. Unfortunately, most useful tools for end-user (ConExp and ToscanaJ) did not have official updates from 2006.

The main problem of compared tools is low limits of size of interactively analyzed artifacts (for example, lattices with more than 8000 concepts can hardly be operated and visualized on modern hardware). This is mainly due to the use of Java and cross-platform GUI or different goals of developing. The current version of Cordiet-FCA can manipulate bigger lattices. After all planned optimizations we will present comparison of implementations of all basic algorithms in the form of compiled components and scripts (Cordiet-DMS platform has built-in tools for benchmarking).

5 Conclusion and future work

Cordiet-DMS is a powerful platform for developing applied software tools, for example, Cordiet-FCA for analyzing data with FCA. This analysis can give us insights into underlying conceptual structure of the data. For the dynamic text collections we can prepare several profiles and iteratively check the sequence of concept lattices.

We assume to improve methodology, extend the set of solvers, optimize some algorithms and use proposed system in different data mining tasks. Some of new solvers will be based on concept stability [5] and similarity [6] calculation algorithms. Also we will extend our platform with triadic concept analysis and noise-robust triclustering methods [7]. Also brand new lattice visualization technique is almost

done with antialiasing, scaling, iceberg concept lattices drawing and more. The next major release of the software (1.0) is planned for November 2012.

It's important to us to provide a freeware version of Cordiet-FCA, that can be extended by community and used in various application fields.

Acknowledgements

The results of the project "Mathematical Models, Algorithms, and Software Tools for Intelligent Analysis of Structural and Textual Data", carried out within the framework of the Basic Research Program at the National Research University Higher School of Economics in 2012, are presented in this work.

Jonas Poelmans is an aspirant at the Research Foundation Flanders.

References

1. Ganter B., Wille R. Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
2. Poelmans J., Elzinga P, Neznanov A., Viaene S., Kuznetsov S., Ignatov D., Dedene G. Concept Relation Discovery and Innovation Enabling Technology (CORDIET) // CEUR Workshop proceedings Vol-757, CDUD'11 – Concept Discovery in Unstructured Data, 2011.
3. Grange E. DelphiWebScript Project (http://delphitools.info/dwscript)
4. Apache Lucene (http://lucene.apache.org)
5. Kuznetsov S.O. On Stability of a Formal Concept // Annals of Mathematics and Artificial Intelligence, Vol. 49, pp.101-115, 2007.
6. Klimushkin M.A., Obiedkov S., Roth C. Approaches to the Selection of Relevant Concepts in the Case of Noisy Data // 8th International Conference on Formal Concept Analysis (ICFCA 2010),. pp. 255-266, 2010.
7. Ignatov D.I., Kuznetsov S.O., Magizov R.A., Zhukov L.E. From Triconcepts to Triclusters // Proceedings of 13th International Conference on rough sets, fuzzy sets, data mining and granular computing (RSFDGrC 2011), LNCS/LNAI Volume 6743/2011, Springer, pp. 257-264, 2011.